

POLITECNICO DI TORINO

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

Tesi di Laurea

VirtualExpo: da esperienza interattiva a metaverso grazie a WebRTC



Relatori

Prof. Andrea Bottino
Dott. Francesco Strada
firma dei relatori

.....
.....

Candidato

Lorenzo Luigi Barbaglia
firma del candidato

.....

Anno Accademico 2021-2022

Sommario

Questa tesi si basa sul tirocinio ed il lavoro svolto per un periodo di circa 300 ore in Funix dove ho avuto la possibilità di lavorare ad un progetto di realtà virtuale interattiva, sviluppato in WebGL, con lo scopo di avvicinare mondi virtuali e tridimensionali a realtà aziendali non solo tramite mondi fittizi ma anche tramite una comunicazione sociale dei futuri utenti, creando quindi un luogo in cui collima il concetto di metaverso. Si estrapolerà il concetto di metaverso, ovvero una rete di mondi virtuali interconnessi con le reti sociali, partendo dall'idea e definendo alcuni esempi che riescono a darci un'impressione di ciò che oggi è il metaverso, essendo un trend sempre più in evoluzione alimentato da sviluppi tecnologici e dal desiderio umano per nuove esperienze. Il concetto di metaverso punta ad un'evoluzione di Internet (Web 3.0) e degli ecosistemi digitali. Punta ad uno spazio virtuale e immersivo dove le persone potranno interagire, lavorare, comprare beni e conoscere realtà in un modo diverso ed innovativo. Grazie ai massicci investimenti da parte dei colossi tecnologici, la creazione di uno spazio virtuale risulta sempre di più un attrattore di modernità, progetti unici e di opportunità per le aziende. In un contesto come quello attuale l'azienda Funix si propone di standardizzare e rendere più accessibile una rappresentazione del metaverso con la piattaforma VirtualExpo. Tramite il Game Engine Unity e la libreria grafica WebGL, è stato creato un progetto base che permette a tutte le aziende interessate, di avere un room tour virtuale di ciò che più gli interessa. In base alle richieste dei compratori finali viene creato un ambiente che poi potrà essere arricchito e gestito dal cliente in base alle sue necessità tramite una Dashboard. Si vuole quindi creare una simulazione che sia capace di sensibilizzare e rendere realtà aziendali più vicine, consentendo agli utenti che ne usufruiranno di essere parte attiva nella scoperta e nell'avvicinarsi a mondi che riteneva lontani senza creare disparità. Grazie alla tecnologia open source di WebRTC è inoltre possibile interagire creando una rete di comunicazione tra più persone all'interno degli ambienti di VirtualExpo. Si potrà comunicare tramite video ottenuti dalla webcam, chat vocale e chat testuale. Si vuole rendere l'esperienza di visita condivisibile con altri tramite la comunicazione, rendendola meno solitaria e fine a sé stessa. Per permettere ad ogni persona che partecipa di sentirsi unica e distinguibile quando esplora gli ambienti del progetto VirtualExpo è stata integrata la SDK di Ready-PlayerMe un tool esterno che permette agli utenti di generare il proprio avatar virtuale. Si esploreranno le diverse tecnologie utilizzate cercando di approfondire il lavoro svolto soprattutto sul lato comunicazione tramite WebRTC, la comunicazione tra Unity WebGL con Javascript e l'integrazione di tool esterni, definendo il funzionamento, le prove svolte e quindi le soluzioni adottate.

Indice

Elenco delle figure	5
1 Introduzione generale	7
1.1 Obiettivo della tesi	7
1.2 Metaverso: Concetto e Definizione nel Mondo Reale	7
1.3 Funix	10
2 Stato dell'arte	11
2.1 Metaverso e Realizzazione	11
2.2 The SandBox - Luogo di incontro nel Gaming	11
2.2.1 Economia	12
2.2.2 Marketing	14
2.3 Fortnite-La collisione di più universi	15
2.4 Spot-Virtual Office	16
2.4.1 Servizi	16
2.4.2 Creazione Stanze	16
2.4.3 Canali Vocali e Video	17
2.4.4 Avatar e Comunicazione nel Virtual Office	18
2.5 Spatial - Gallerie d'arte digitali	19
2.5.1 Home	19
2.5.2 NFT	20
2.5.3 Comunicazione Creazione di Contenuti	20
2.5.4 Personalizzazione Avatar	21
2.5.5 Creare una stanza	21
2.5.6 Portali	23
2.6 Virtual Expo	24
2.6.1 Dashboard	24
2.6.2 Front-End	31
3 Metodo	35
3.1 Task Principali	35
3.2 Avatar	36
3.2.1 Prima e Terza Persona	36
3.2.2 Movimento Avatar	39

3.2.3	Cambio Avatar - ReadyPlayerMe	42
3.3	Multiutente	48
3.3.1	WebRTC Tecnologia e Funzionamento	48
3.3.2	SkyWay Implementazione WebRTC nel progetto	50
3.4	Comunicazione	56
3.4.1	Media in WebRTC	56
3.4.2	Skyway e Condivisione video e audio tra utenti	57
3.4.3	Condivsione Video	59
4	User Test	61
4.1	Raccolta	61
4.2	Test Aziendali Interni	61
4.3	Test Gruppo di Studenti	63
5	Conclusione e Sviluppi futuri	67

Elenco delle figure

2.1	Esempio dello shop in 'The Sandbox'	12
2.2	Esempio dell' Editor in 'The Sandbox'	13
2.3	Mappa del mondo virtuale 'The Sandbox'	14
2.4	Editor Stanze in 'Spot'	16
2.5	Chat vocali in 'Spot'	17
2.6	Editor Avatar ed esempio emoticon in 'Spot'	18
2.7	Tab di navigazione stanze nella home di 'Spatial'	19
2.8	Pagina OpenSea di 'Spatial'	20
2.9	Tab di personalizzazione avatar 'Spatial'	21
2.10	Editor di stanze 'Spatial'	21
2.11	Editor quadri digitali 'Spatial'	22
2.12	Portali per altre stanze 'Spatial'	23
2.13	UI allestimento 'Virtual Expo'	24
2.14	Inspector Oggetto 'Virtual Expo'	25
2.15	Creazione stanza 'Virtual Expo'	26
2.16	Editor Radar 'Virtual Expo'	26
2.17	Lista dei modelli presenti in 'Virtual Expo'	27
2.18	Visualizzazione Orbit 'Virtual Expo'	28
2.19	Lista Hotspot 'Virtual Expo'	29
2.20	Inspector Slideshow 'Virtual Expo'	30
2.21	Vista della pagina di 'Virtual Expo'	31
2.22	Navigazione in 'Virtual Expo'	32
2.23	Menu laterale in 'Virtual Expo'	32
3.1	Avatar nella sezione di Test	36
3.2	Diagramma script cambio POV	37
3.3	Effetto di dither sull'avatar	37
3.4	Diagramma script cambio shader	38
3.5	Curva applicata allo shader	38
3.6	Navmesh su cui si muovono i NavMeshAgent	40
3.7	Diagramma Script Movimento tramite input direzionali	40
3.8	Diagramma Script Movimento tramite input click del mouse	41
3.9	Diagramma Script coroutine	42
3.10	Editor avatar Ready Player Me	43
3.11	Esempio URL univoco fornito	44

3.12	Pacchetti plugin RPM in Unity	44
3.13	Tab RPM in Unity	45
3.14	Animator Avatar	46
3.15	Formola calcolo velocità	46
3.16	Diagramma delle processo di download ed istanziamento di un Avatar da RPM	47
3.17	WebRTC Logo	48
3.18	SkyWay Logo	50
3.19	Comunicazione con il Signaling Server	51
3.20	Comunicazione con il STUN Server	51
3.21	Comunicazione con il TURN Server	52
3.22	Cofigurazione SFU e Mesh dei peer	52
3.23	Diagramma delle funzioni di Setup	53
3.24	Diagramma di un HTML event che manda un comando ad altri utenti	54
3.25	Diagramma di ciò che accade non appena un nuovo user entra in una stanza	55
3.26	Diagramma di ciò che accade quando un Peer si collega	57
3.27	Valori funzionamento della stanza	57
3.28	Tag video personale e degli altri utenti connessi alla stanza	58
3.29	Diagramma di ciò che accade quando un Peer si scollega	58
3.30	Diagramma degli eventi di condivisione schermo	59
3.31	Condivisione del video in scena	59

Capitolo 1

Introduzione generale

1.1 Obiettivo della tesi

Scopo del progetto Virtual Expo di Funix è quello di ottenere un pacchetto in Unity che possa essere riutilizzabile mantenendo la struttura di base e cambiando solamente gli asset, per creare ambienti visitabili e interagibili in diversi modi con il supporto di una chat video e audio per permettere l'interazione tra più utenti. Quindi l'obiettivo di questa tesi è definire l'industria che si basa sul concetto di metaverso, il suo significato e le sue implicazioni nel futuro delle tecnologie, ma soprattutto il mio percorso all'interno dell'azienda, le parti a cui ho lavorato e le tecnologie utilizzate per il progetto. In particolare lo sviluppo di tutto ciò che riguarda l'utente quindi la creazione di un avatar in tutte le sue parti dal sistema di movimento alla personalizzazione tramite SDK esterna, la realizzazione di un supporto di comunicazione video e audio da integrare alla piattaforma 'Virtual Expo' tramite la tecnologia open source di WebRTC. Sarà quindi testata con diversi partecipanti l'usabilità della piattaforma, del movimento nell'ambiente e soprattutto la funzionalità della comunicazione tra i partecipanti. Il canale di comunicazione si rivela essenziale all'interno del progetto questo perché rende i futuri progetti commissionati, un punto di incontro nei quali è possibile comunicare e partecipare attivamente negli ambienti virtuali con più persone e non più in solitaria in maniera passiva. Inoltre l'implementazione tecnica di queste feature evidenzia la capacità dell'azienda di affrontare la sfida di creare e replicare ambienti con un'impronta al metaverso sempre più richiesti dalle aziende che credono in una comunicazione innovativa e differente.

1.2 Metaverso: Concetto e Definizione nel Mondo Reale

Nella letteratura il termine Metaverso viene coniato da Neal Stephenson, nel romanzo di fantascienza, Snow Crash, nel romanzo [Stephenson \[1992\]](#) indica il metaverso come un mondo virtuale nel quale è possibile incontrarsi con le persone tramite avatar in uno spazio libero, con locali e negozi creato da programmatori indipendenti. Dalla definizione

data dall'autore si possono definire alcuni punti fondamentali che definiscono il concetto ovvero:

- Presenza: quindi la sensazione di essere in uno stesso luogo virtuale nonostante gli utenti si trovino in luoghi separati
- Interoperabilità: ovvero la possibilità di muoversi tra diversi ambienti virtuali mantenendo le informazioni permettendo di utilizzarle (per esempio l'avatar)
- Standard: quindi un metodo comune per permettere la comunicazione, spendere e navigare in un mondo interoperabile, serve quindi un'infrastruttura che sia capace di reggere, sono quindi necessari standard tecnologici comuni e decentralizzati.

Il concetto di metaverso affascina per la potenza che rappresenta tuttavia oggi risulta ancora molto lontano dalla realizzazione se non impossibile, questo perché manca la struttura stessa sulla quale si dovrebbe creare questo 'nuovo mondo' ha regole precise e non compatibili con quelle del concetto di metaverso.

Spesso accompagnata alla parola metaverso si associa la nuova evoluzione del web che passa da Web 2.0 a Web 3.0 [Nath \[2022\]](#). Si vuole quindi sottolineare il passaggio da Web 2.0 con cui si definisce l'era dei social media, delle communities e della libertà di espressione, ovvero un web segnato dalla partecipazione, che negli ultimi anni ha rivelato i suoi limiti e rischi quali privacy, sicurezza e mercato predominato da pochissime aziende le cosiddette GAFAM, sigla che rappresenta le più importanti aziende dell'IT (Information Technology) in particolare Google (Alphabet), Apple, Facebook (Meta), Amazon e Microsoft. Da queste problematiche nasce l'idea di Web 3.0, basata su un internet più sicuro, corretto e trasparente costituito da una rete aperta e decentralizzata, quindi un web che punta ad una rete più incentrata sull'uomo dove ognuno può accedere ai contenuti liberamente e in sicurezza. Ed è qui che si incastra il concetto di metaverso all'idea di Web 3.0 siccome il Web3.0 riesce a realizzare l'idea di un internet decentralizzato più Human Centered una rete dentro la quale gli utenti tramite meccanismi di blockchain permettono di impiegare criptovalute per acquistare beni e servizi in ambienti virtuali. Tuttavia questa connessione non è priva di criticità date principalmente dall'idea di web3.0 la quale necessita di ristrutturare totalmente la rete che al suo stato attuale è un'idea pressoché impraticabile e per nulla sostenibile (anche dal punto di vista della sostenibilità ambientale e sociale), lo scenario migliore è quello di un'integrazione di queste realtà tuttavia non rendendo praticabile una decentralizzazione totale vista la forza e le risorse è difficile immaginarsi un nuovo web non sostenuto e governato dai GAFAM. Ciononostante il metaverso è la prossima grande innovazione tecnologica, che sempre di più attira produttori di giochi online, social network e aziende leader della tecnologia per conquistare una fetta di questo mercato che è in crescita costante e che già oggi è ricco di investimenti e coinvolgimento da parte delle aziende e del pubblico [[Svend et al. \[2022\]](#)]. Il concetto di metaverso è un forte attrattore questo perché le aziende cercano sempre di più di accorciare la distanza tra di loro e l'utenza a cui si rivolgono: l'idea di potersi vedere all'interno di un ambiente, confrontare con il pubblico ed interagire con l'ambiente crea un rapporto emotivo più forte tra azienda e utente siccome si incentiva una partecipazione più attiva e coinvolgente. Quindi oggi giorno il metaverso è nulla più che un'idea, un concetto che

permette alle aziende di sognare, impegnarsi verso un utilizzo di internet diverso e nuovo e soprattutto il concetto essendo un grande attrattore aiuta a vendere meglio le proprie tecnologie.

Le tecnologie essenziali per la realizzazione del metaverso sono quelle di Virtual Reality, visori, e di Augmented Reality, visori che non occludono totalmente la vista. Sono tecnologie che permettono alle persone di immedesimarsi in un mondo virtuale tramite una simulazione del mondo fisico dove le persone possono comunicare, lavorare, studiare, giocare e tante altre cose tramite i propri avatar virtuali. Le tecnologie AR/VR saranno punti cardini del metaverso, perché in grado di fornire un'esperienza unica ed immersiva, le tecnologie VR saranno in grado di fornire un'esperienza sensoriale più profonda permettendo di avere un contatto più vivido con i mondi virtuali, mentre le tecnologie AR permetteranno un'immersività maggiore per esempio con gli oggetti del mondo reale combinati a quello virtuale. Essendo che il metaverso promette di fornire un'esperienza totalmente immersiva e dinamica in ambienti 3D, le tecnologie VR permettono una telepresenza reale, la vera sfida per queste tecnologie sarà quella di rendere più sfocata le azioni che ad oggi sono distinte tra vita virtuale e vita reale, come per esempio il lavoro da remoto. Spesso si confondono i termini Virtual Reality e Metaverso, rendendoli intercambiabili ma se la VR permette una telepresenza permettendo una comunicazione più semplice con il mondo virtuale con Metaverso si intende un concetto più ampio ovvero la possibilità di connettere le persone da tutto il mondo attraverso un mondo virtuale, condiviso e multifunzionale, mutando quindi il modo in cui le persone si collegano tra di loro creando un ponte tra mondo fisico e virtuale garantendo al contempo trasparenza e sicurezza. Tramite la Augmented Reality è possibile fornire agli utenti un'estensione del mondo fisico tramite quello virtuale, grazie a tecnologie che permettono di effettivamente vedere il mondo reale al quale si possono apportare modifiche virtuali grazie a smartphones o smart glasses, quindi chi propone un metaverso basato sulle tecnologie AR ritiene che si formerà una sintesi tra mondo reale e virtuale. [Timothy and M. Claudia [2018]]

Dopo gli annunci di Facebook di puntare la futura crescita aziendale nel metaverso, tramite la piattaforma Horizon Worlds ed in tecnologie sofisticate per l'immersione, cambiando il nome dell'azienda stessa in Meta ad indicare le intenzioni precise della compagnia, si è creato un movimento all'interno del mercato che sempre più vede la possibilità di condividere un ambiente virtuale con una community. Certamente i GAFAM nel futuro prossimo cercheranno di strappare una parte di questo mercato nonostante i loro progetti siano vaghi, sia dal punto di vista teorico che tecnico. Probabilmente l'emersione del metaverso da idea aziendale a luogo di incontro del pubblico potrebbe risultare lenta proprio per i diversi investimenti aziendali che spezzano e diversificano le tecnologie. Meta ha acquisito Oculus ottenendo il leader della tecnologia VR, Virtual Reality, consumer, Google invece si piazza nel mercato AR, Augmented Reality, con Magic Leap occhiali capaci di unire il mondo virtuale a quello reale e così anche Microsoft tramite Hololens. Così facendo queste colossi tecnologici si sforzano di creare un ecosistema hardware che si combinerà a quello dei loro servizi. Sembra quindi che il mercato nonostante gli investimenti sia ancora tiepido nello sbilanciarsi totalmente verso la virtual reality e quindi un metaverso persistente piuttosto che la augmented reality cercando quindi di estendere la realtà. Certamente questo tipo di investimenti sulle tecnologie che permettono un maggior coinvolgimento dell'utente saranno i blocchi costitutivi per il metaverso che vede la

sua realizzazione in nicchie promettenti come gli uffici virtuali che permettono un'elasticità nella vita delle persone, oppure nell'educazione permettendo quindi un coinvolgimento maggiore delle classi che potranno esperienziare in maniera più viva le lezioni, campo dove certamente il metaverso e le tecnologie annesse avranno modo di espandersi e realizzarsi è quello del gaming permettendo un coinvolgimento più intenso e vivido delle esperienze, altro campo nel quale si stanno formando nuove opportunità è quello dell'e-commerce che trasforma i soliti cataloghi statici in esperienze interattive, fornendo la possibilità ai consumatori di esplorare i negozi ed i loro prodotti direttamente da casa. Quindi lo sviluppo del metaverso è una sfida in corso che richiede un'infrastruttura solida e sicura, l'interoperabilità è la questione chiave che potrebbe frenare l'implementazione del metaverso. Per il concetto che esprime per esempio un utente dovrebbe essere capace di fare spese nel metaverso attraverso tutto l'ecosistema, non solo tramite un'applicazione stand alone. Per questo i giganti tech stanno facendo investimenti importanti in visione di un mercato che si potrà realizzare anche tramite le tecnologie quali VR e AR che permettono di avere condizioni favorevoli alla nascita del metaverso o applicazioni simili.

1.3 Funix

Funix nasce all'inizio del 2015 all'interno dell'incubatore del Politecnico di Torino I3P, qui sviluppano esperienza professionale e una rete di contatti lavorativi che gli permetterà di intraprendere il proprio percorso in maniera indipendente. Come visione aziendale Funix punta a sviluppare soluzioni innovative, dinamiche e personalizzate. Senza però dimenticare l'usabilità siccome sì le tecnologie avanzano e diventano sempre più alla portata di chiunque, ma proprio per la loro rapidità i paradigmi di funzionamento cambiano di continuo rendendo necessario uno studio sull'intuitività e l'usabilità dei prodotti. La missione diventa quindi quella di avvicinare la tecnologia all'individuo in quanto le innovazioni tecnologiche sono essenziali per il miglioramento della vita delle persone. I servizi che Funix è in grado di offrire ai propri clienti riguardano la Real Time Interactive Visualization quindi la progettazione e realizzazione di visualizzatori e configuratori di prodotti interattivi per ogni piattaforma. Si occupano inoltre di Virtual and Augmented Reality per cui la realizzazione di software interattivi con il focus sulla realtà aumentata e la realtà virtuale, con la possibilità di creare piccole piattaforme web standalone o a supporto di applicativi prodotti. Ovviamente tutte queste realtà e mondi virtuali devono essere supportati dalla creazione di materiale grafico per permettere l'interazione in ogni piattaforma, unita ad una approfondita conoscenza della grafica 3D quindi modellazione e rendering. Il ricco portfolio aziendale descrive appieno le capacità di chi programma, modella e studia l'esperienza utente all'interno di Funix. Esperienze in VR, video renderizzati, installazioni per fiere e pagine in WebGL sono alcune delle tante possibili scelte di cui appunto è possibile visionare e saggiare la qualità attraverso il loro portfolio o tramite la piattaforma di Virtual Expo che permette di esplorare un ambiente virtuale e osservare non solo le capacità della piattaforma ma anche i diversi lavori svolti negli anni dall'azienda.

Capitolo 2

Stato dell'arte

2.1 Metaverso e Realizzazione

Tralasciando le grandi aziende che muovono i mercati ed il futuro delle tecnologie, esistono diverse realtà che propongono la loro visione di quello che è il metaverso come esperienza continua che integra diverse piattaforme e porta le persone in luogo virtuale in cui incontrarsi piuttosto che estendere la realtà tramite l'utilizzo di tecnologie ad hoc.

Il metaverso, per come si realizza il concetto nel concreto, fonde senza soluzione di continuità giochi, realtà virtuale, live streaming, criptovalute e social media, mentre gli utenti si muovono attraverso un ecosistema di prodotti concorrenti. Inoltre le connessioni internet sempre più veloci permettono di vivere l'esperienza online senza pause o reset.

2.2 The SandBox - Luogo di incontro nel Gaming

Esempio di piattaforma legata al gaming è 'The Sandbox' si tratta a tutti gli effetti di un gioco in realtà virtuale [Borget [2021]]. La piattaforma è basata su Ethereum, piattaforma decentralizzata del Web 3.0 per la pubblicazione di Smart Contracts, e nel mondo di gioco gli utenti ed il mondo sono rappresentati tramite una grafica in Voxel-Art. In questa piattaforma gli utenti diventano proprietari di quello che creano tramite blockchain e smart contracts ovvero protocolli di transazione con lo scopo di eseguire, manipolare o segnalare automaticamente attività legalmente applicabili, riducendo la necessità di intermediari. Viene definito come Metaverso decentralizzato dove quindi non è la software house a decretare il valore dei beni ma sono gli utenti stessi che popolano il mondo di gioco. In questo mondo virtuale si ribalta l'idea di "Free to Play", giochi gratuiti spesso accompagnati dalla possibilità di microtransazioni, a "Play to Earn" concetto basato sul fatto che ogni giocatore è libero di guadagnare creando contenuti e oggetti.

2.2.1 Economia

I token di 'The SandBox', ovvero informazioni digitali all'interno di una blockchain che conferiscono diritti ad un determinato utente, permettono di accedere a beni, spazi e servizi per acquistare terreni sulla mappa, vengono definiti LAND i quali sono limitati a 166464. Si tratta di spazi con una dimensione definita di Voxel (pixel tridimensionali). Per acquistare i diversi beni si utilizza la criptovaluta SAND grazie alla quale sarà possibile svolgere tutte le attività all'interno del mondo di gioco come organizzare eventi, costruire edifici, affittare spazi e altro.

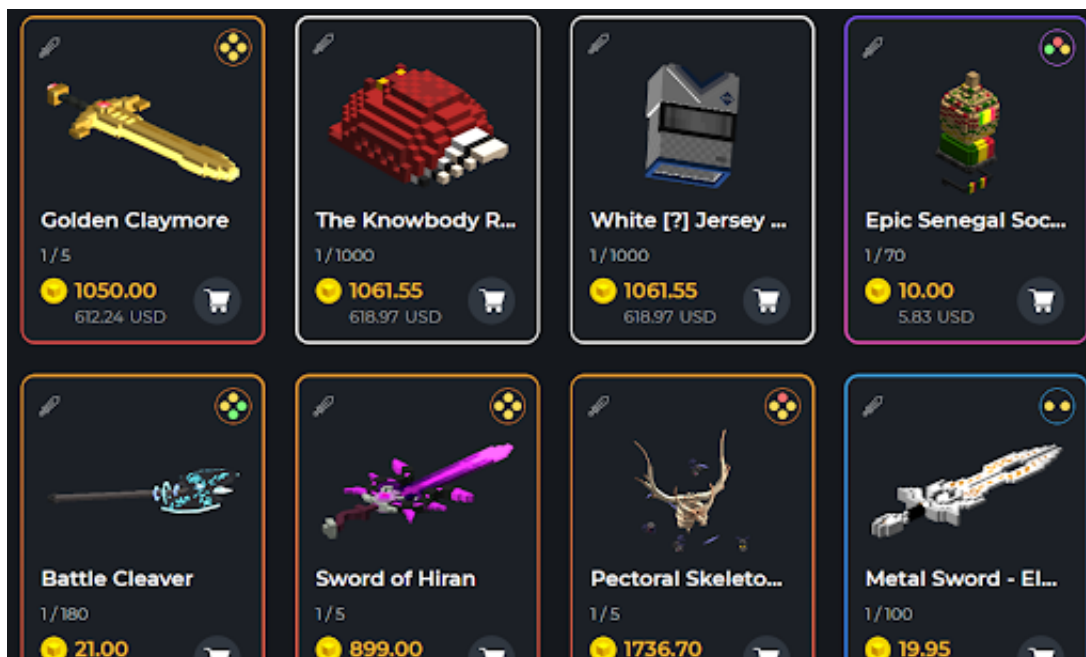


Figura 2.1. Esempio dello shop in 'The Sandbox'

Gli utenti possono quindi creare i propri asset ovvero elementi utilizzabili nel mondo quali vestiti, oggetti di qualsiasi genere o edifici. Per creare oggetti in 'The Sandbox' è disponibile un editor chiamato VoxEdit che permette di creare asset che si potranno importare nel mondo virtuale oppure nel marketplace dove è possibile vendere le proprie creazioni. Un altro modo di guadagnare oltre alla creazione di Asset o l'acquisto di Land è tramite il Game Maker proprietario gratuito, uno strumento che non richiede doti di programmazione che permette di creare giochi che si possono vendere sul marketplace.

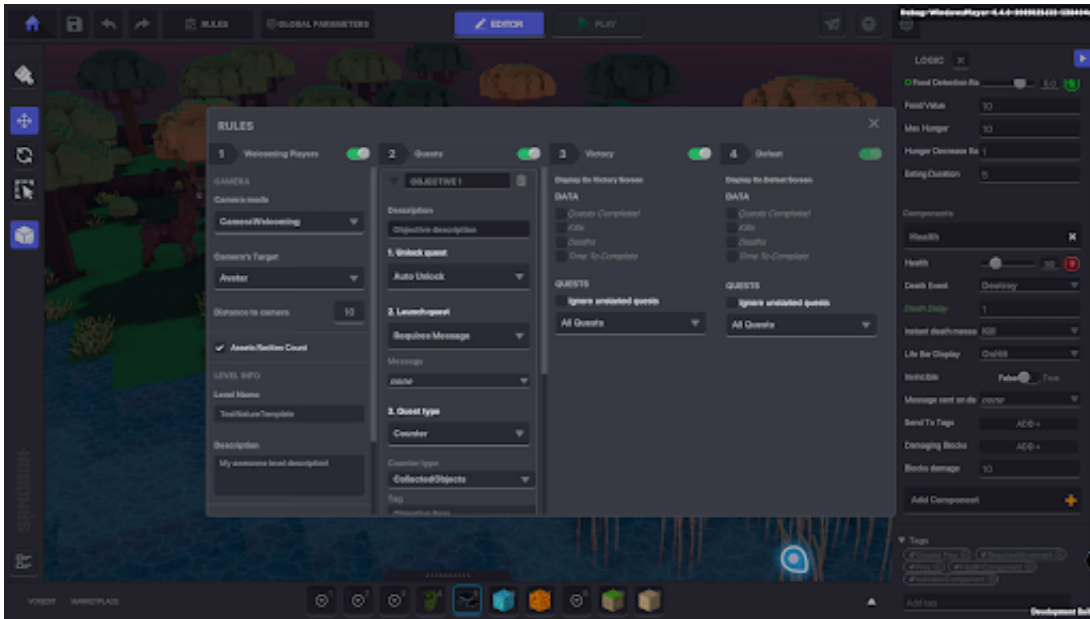


Figura 2.2. Esempio dell' Editor in 'The Sandbox'

2.2.2 Marketing

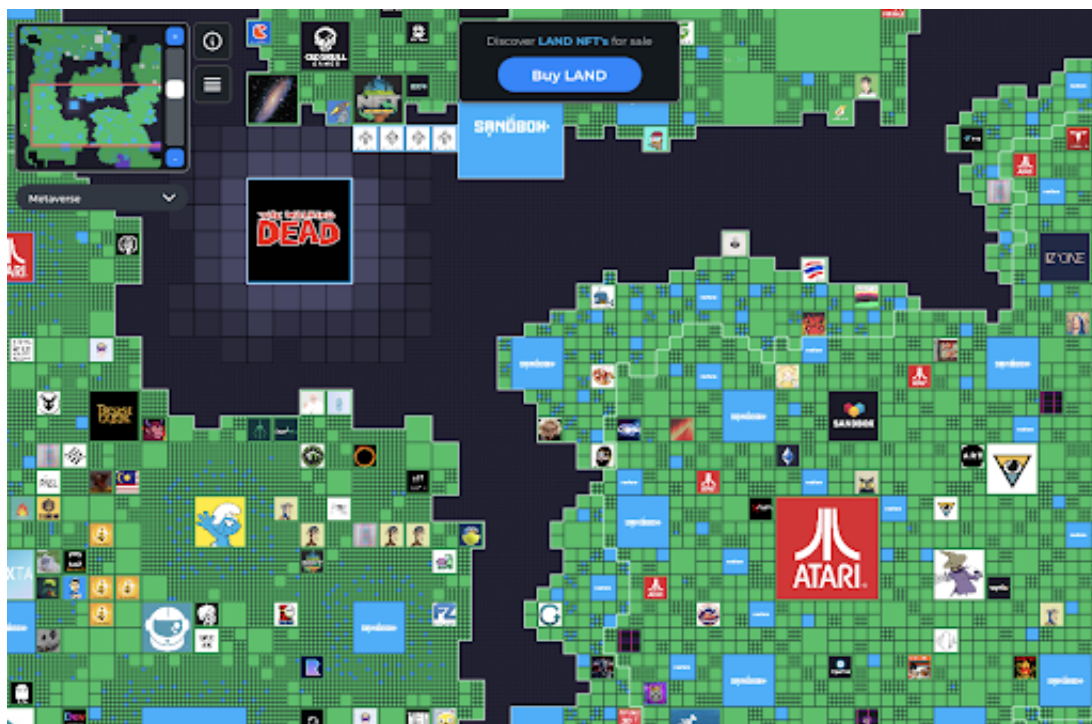


Figura 2.3. Mappa del mondo virtuale 'The Sandbox'

Tra i diversi proprietari di terreni ci sono aziende riconoscibile e famose quali Atari, che ha ricreato i giochi storici nei suoi terreni, oppure è presente The Walking Dead che ha realizzato un gioco con i suoi personaggi oppure sono presenti aziende del mondo crypto come Binance, il giornale The China Morning o brand di vestiti come Adidas. Avere un terreno da diritto di creare mondi digitali nei quali è possibile ospitare evento, mostre, giochi, concerti o semplici luoghi di incontro con possibilità di renderli a pagamento. Inoltre le aziende possono rendere disponibili capi di abbigliamento, oggetti o altro sotto forma di NFT in modo da promuoversi all'interno del metaverso, permettendo una fidelizzazione del pubblico. È chiaro che oggi chi sceglie di essere su The SandBox la fa per la visibilità e per intercettare un pubblico specifico difficilmente raggiungibile. Non è ovviamente una scelta economica in termini monetari ma anche di risorse e tempo da dedicare siccome richiede una cura della propria presenza definendo quindi un nuovo modo di fare marketing tramite luoghi virtuali.

2.3 Fortnite-La collisione di più universi

Esempio eclatante nel gaming è senza dubbio Fortnite [Tassi [2022]], il quale è riuscito ad abbattere le barriere tra diversi universi della cultura pop mescolandoli in un unico ambiente condiviso. Fortnite nasce nel 2017, sviluppato da ‘People Can Fly’ e pubblicato da Epic Games, presenta 3 diverse modalità di gioco ma senza dubbio il successo nasce dalla modalità Battaglia Reale, nella quale 100 giocatori si scontrano per la sopravvivenza. Nonostante il gameplay da sparatutto in terza persona, Fortnite negli anni è sempre più riuscito ad ottenere contratti con diversi brand riuscendo a generare un mondo virtuale nel quale il crossover tra realtà diverse diventa la normalità sarà quindi possibile vedere Goku di Dragon Ball scontrarsi con Darth Vader di Star Wars, e così per tantissimi altri brand ma non solo infatti è importante evidenziare la ‘Icon Series’ ovvero una tipologia di skin e oggetti estetici utilizzabili nel gioco, ottenibili tramite pacchetti. Questa serie è quel passo verso un metaverso nel quale realtà e finzione si uniscono infatti è possibile ottenere skin di artisti quali Ariana Grande o Bruno Mars, personalità dello sport come LeBron James o content creator quali Ninja o LazarBeam, sicuramente la collezione non conta un’infinità di nomi tuttavia sottolinea l’interesse da parte di Fortnite di attrarre sempre più persone nel suo mondo tramite personalità o brand familiari all’utente. Si crea quindi una commistione di mondi diversi e spesso opposti in Fortnite che tuttavia non stona perché perfettamente coerente con l’universo creato dagli sviluppatori e raccontato durante gli anni . Nonostante le collaborazioni con personalità e brand conosciuti ed esaltati dalla cultura pop, il vero potenziale di Fortnite come metaluogo di incontro non nasce dalle assurde battaglie tra personaggi improbabili, ma dagli eventi in partnership con Artisti come la già citata Ariana Grande oppure Travis Scott. In questi eventi unici live a cui è possibile partecipare per un tempo limitato a determinati orari si partecipa a veri e propri concerti virtuali nei quali si rivoluziona l’esperienza musicale tramite momenti in cui l’utente stesso è coinvolto e altri nei quali personificazioni degli artisti cantano i loro brani tramite coreografie assurde e spettacolari. Il potenziale e poi l’effettiva potenza di questi eventi è evidente soprattutto per la partecipazione massiva degli utenti contando milioni di spettatori, che partecipano ad un’esperienza unica e coinvolgente evidenziando ancor di più la capacità di Fortnite di creare un metaverso unico. Viene citato Fortnite come caso di studio perché grazie alla lungimiranza e capacità di far coesistere diversi mondi in un unico ambiente riesce a definire quello che le diverse realtà che puntano a creare un metaverso spesso falliscono, ovvero il coinvolgimento del pubblico non solo grazie ad eventi unici ma anche tramite attrattori culturali pop.

2.4 Spot-Virtual Office

Spot [Wes and Gordon [2022]] si tratta di una piattaforma di Virtual Office nasce quindi con lo scopo di trasformare il modo in cui le persone si collegano per lavorare da remoto. Infatti con spot sarà possibile creare il proprio spazio virtuale in base alle necessità.

2.4.1 Servizi

Principalmente Spot si rivolge a gruppi aziendali o ad organizzatori di eventi online, i servizi offerti fornisce ambienti virtuali customizzabili dentro i quali è possibile muoversi, interagire con l'ambiente e comunicare tramite canale video e audio. Spot distingue gli ambienti acquistabili in base al numero di utenti che una stanza può ospitare ed in base alla tipologia per cui verrà usato lo spazio infatti si fa distinzione tra Virtual Offices e Virtual Events, che si distinguono principalmente per la presenza o meno di tool di moderazione presenti nell'opzione eventi virtuali.

2.4.2 Creazione Stanze

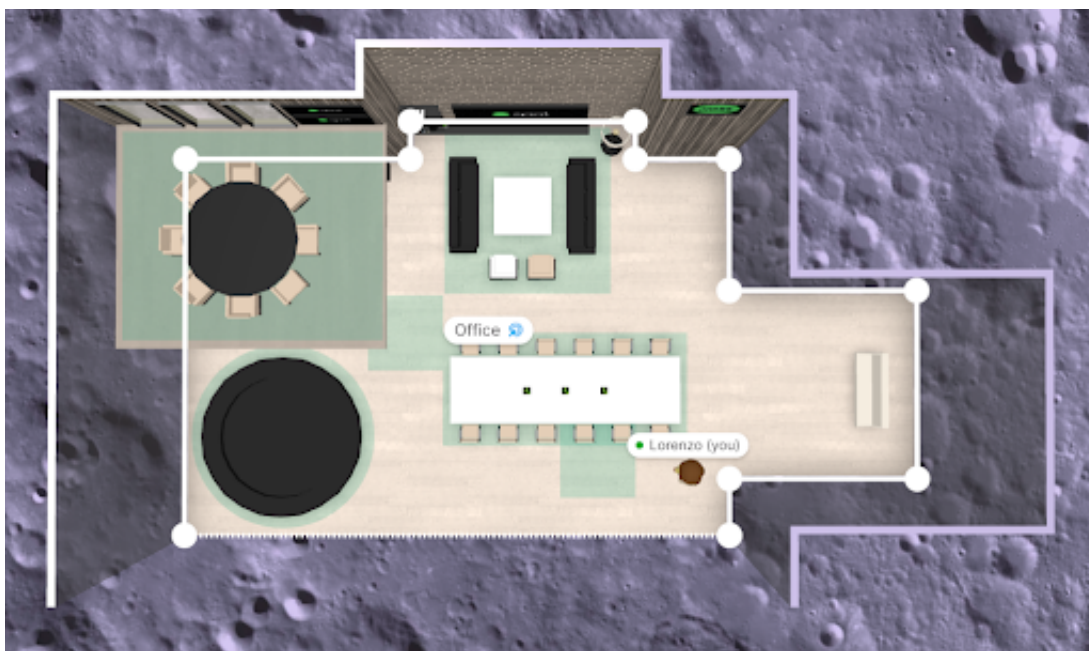


Figura 2.4. Editor Stanze in 'Spot'

Una volta aperto un ambiente scelto tra alcuni template forniti da Spot permette tramite un Build Mode, tool interno, di gestire l'ambiente virtuale in tutte le sue parti dalla pianta dell'ufficio virtuale, nell'arredamento e nei contenuti multimediali. In questa modalità si

avranno a disposizione diversi asset ambientali i quali potranno essere piazzati a piacimento all'interno dell'ambiente, che come detto precedentemente può essere modificato anche nella pianta. Gli asset ambientali sono divisi per categorie e nonostante alcuni siano disponibili solamente sotto pagamento del servizio sono diversi e variegati. Inoltre ogni oggetto può essere personalizzato nei colori che compongono il modello.

2.4.3 Canali Vocali e Video

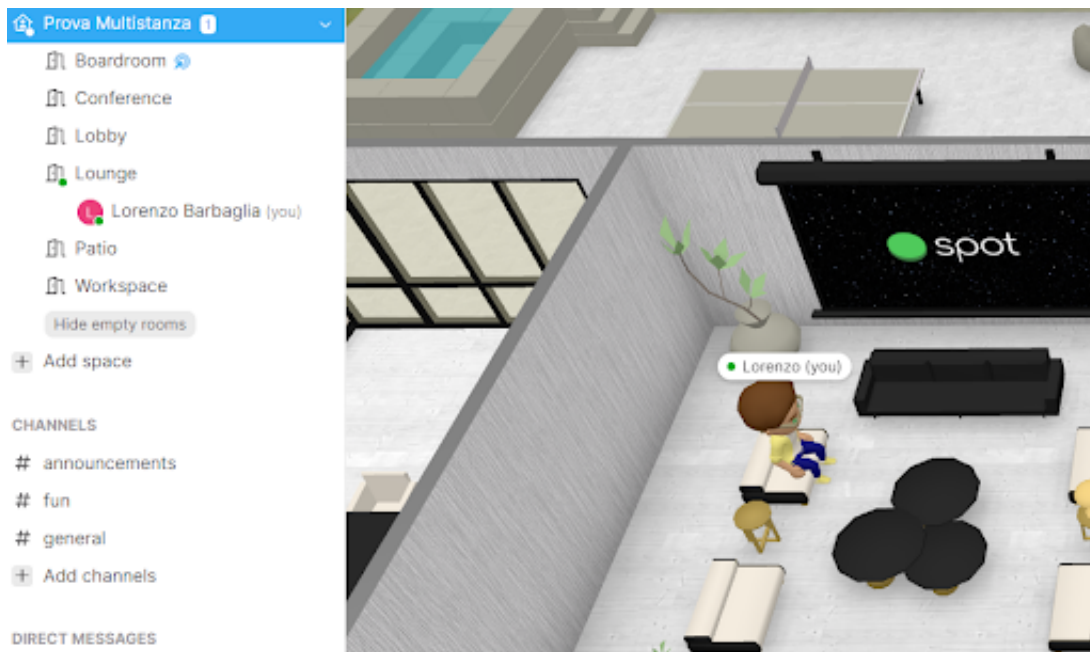


Figura 2.5. Chat vocali in 'Spot'

Oltre alla possibilità di modificare l'ambiente si potranno gestire i diversi spazi dell'ambiente virtuale definendo delle aree che identificano dei canali e quindi diversi canali vocali associati così da avere delle vere e proprie stanze virtuali per ogni esigenza.

2.4.4 Avatar e Comunicazione nel Virtual Office

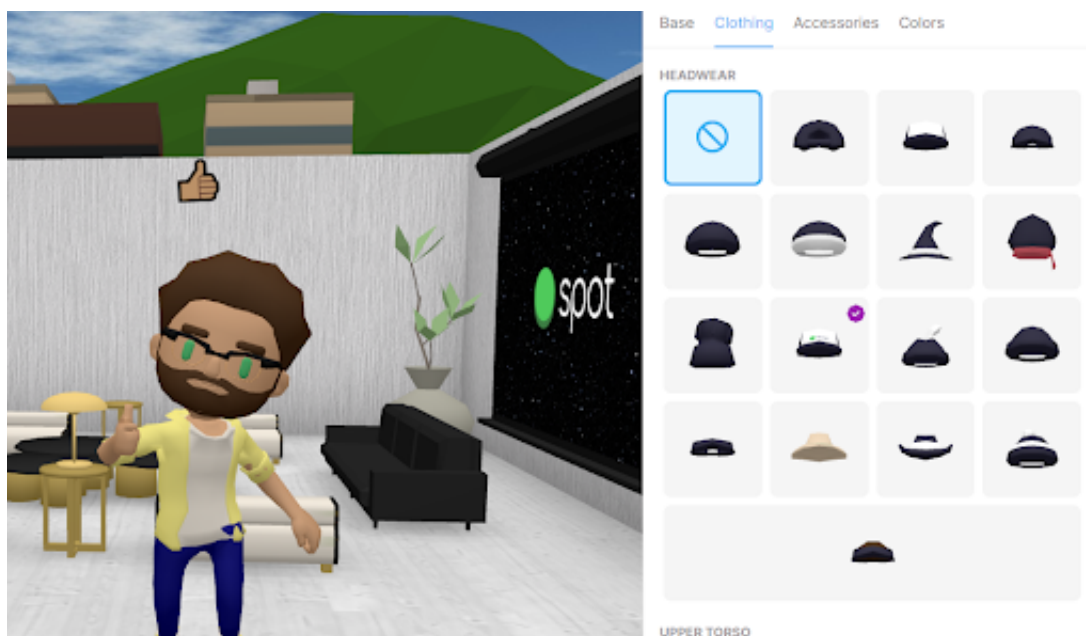


Figura 2.6. Editor Avatar ed esempio emoticon in 'Spot'

L'avatar che ci rappresenta potrà essere personalizzato liberamente tramite un tool di personalizzazione interno e con questo saremo liberi di muoverci nell'ambiente tramite una visuale isometrica e di interagire con gli altri tramite emoticon e con animazioni uniche per esprimere i propri stati d'animo o approvazione, questa comunicazione non diretta è molto presente siccome si potranno impostare anche lo stato di impegno come 'Pausa Pranzo' o 'Focus Time', rendendo molto più evidenti le necessità di ogni partecipante. Quando si decide di attivare la webcam il nostro avatar verrà sostituito con una iframe contenente il nostro video, il quale manterrà tutte le possibilità di movimento precedenti.

Quindi per quanto acerba come piattaforma Spot, sembra aver chiaro il suo target di mercato e potenzialmente gli ambienti generabili possono essere un luogo virtuale di incontro in cui interagire con i propri colleghi di lavoro in maniera meno fredda di una classica chiamata online. Diversamente il lato eventi risulta banale e fine a sé stesso siccome le attività che si possono svolgere in questi ambienti sono limitate e gli ambienti per quanto personalizzabili risultano asettici e poco interessanti per un potenziale evento online.

2.5 Spatial - Gallerie d'arte digitali

Spatial [Anand and Jinha [2022]] similmente a Virtual Expo si dedica ad aiutare i creatori ed i marchi a creare i propri spazi personalizzati nel metaverso. Spatial fornisce agli utenti la possibilità di sfruttare i propri spazi per contenuti, costruire una community e promuovere la vendita di opere digitali o prodotti creativi, infatti è possibile creare i propri spazi 3D che possono essere trasformati in NFT così da poterli affittare o vendere ad altri per poter ospitare le proprie collezioni. Tramite la piattaforma di Spatial si vuole creare l'opportunità per i creator, i loro sostenitori e fan di interagire in maniera rapida e veloce.

2.5.1 Home

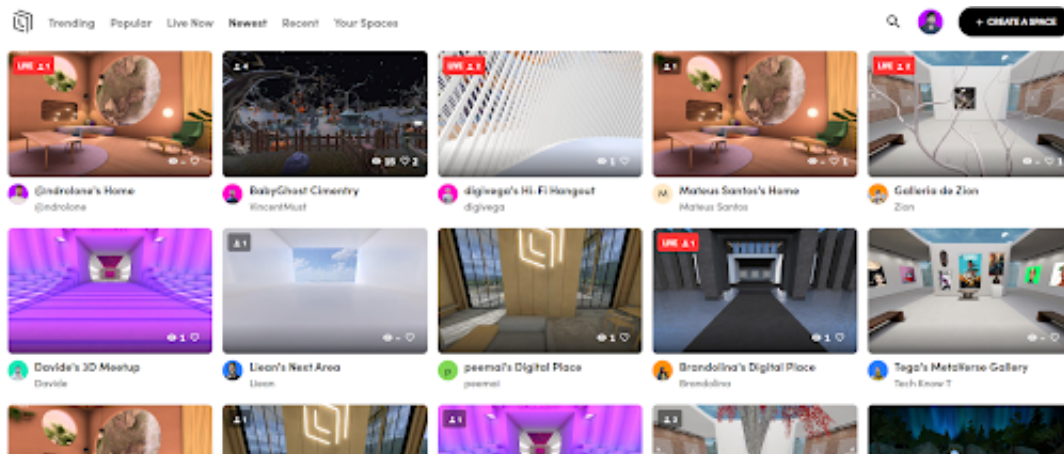


Figura 2.7. Tab di navigazione stanze nella home di 'Spatial'

Una volta arrivati sul sito di Spatial ci si trova di fronte ad una Tab dove è possibile filtrare sia le diverse stanze create dalla community, per stanze di tendenza, popolari o più recenti una volta deciso basta cliccare e si entra nella stanza. Spesso ci si ritrova in ambienti uno uguale all'altro con leggere modifiche, sicuramente è possibile trovare ambienti più complessi e ben studiati tuttavia sono pochi e non coinvolgono la community. Spesso infatti si trovano stanze totalmente vuote o popolate da pochissime persone che non comunicano in alcun modo. Sembra quindi che manchi una direzione precisa da parte della piattaforma per coinvolgere maggiormente le persone nel metaverso che hanno creato.

2.5.2 NFT

La parte più affiatata della community va ritrovata nella community degli NFT siccome spatial è direttamente collegato a OpenSea, marketplace di Non Fungible Token, questo permette di mettere in vendita le proprie stanze per gli utenti ma anche di creare dei musei virtuali dentro i quali esporre le proprie opere digitali come fossero gallerie private, sicuramente è una buona opportunità anche per gli artisti di NFT potendo esporre le proprie creazioni.

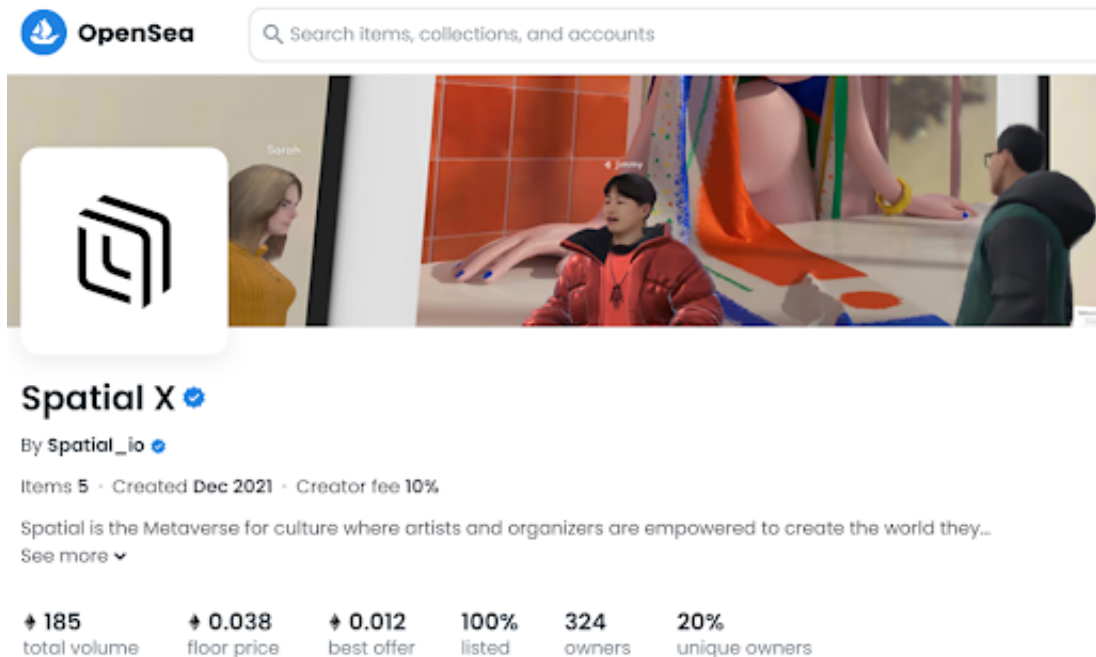


Figura 2.8. Pagina OpenSea di 'Spatial'

2.5.3 Comunicazione Creazione di Contenuti

Dal punto di vista tecnico Spatial ha tutte le feature che ci si potrebbe aspettare da una piattaforma multiutente quindi chat video e audio tramite la quale è possibile interagire con gli altri utenti in maniera personalizzata. Una volta attivato il microfono è infatti possibile filtrare le voci altrui per aver la miglior esperienza. Se si attiva la webcam un cerchio contenente il video verrà mostrato sopra la testa del nostro avatar. Dentro le stanze è possibile tramite un menù ottenere una video registrazione della esperienza all'interno della stanza, creare una foto quindi uno screenshot oppure entrare in 'Filming Mode' che permette di muoversi per la stanza con comandi particolari così da ottenere riprese più cinematografiche rispetto alla semplice registrazione a schermo.

2.5.4 Personalizzazione Avatar



Figura 2.9. Tab di personalizzazione avatar 'Spatial'

Ogni utente potrà creare il proprio avatar dopo aver loggato il proprio profilo tramite l'estensione di Ready Player Me, che permette una personalizzazione più che completa per il proprio utente soprattutto del viso che tramite una foto del proprio volto crea una mesh sulla stessa.

2.5.5 Creare una stanza

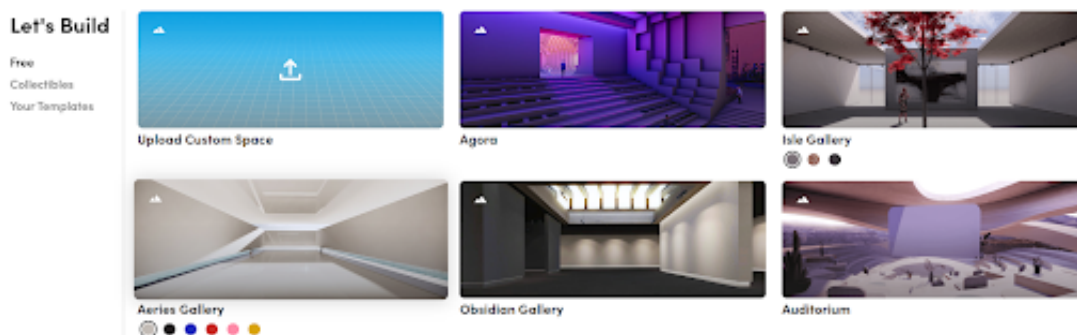


Figura 2.10. Editor di stanze 'Spatial'

In Spatial è possibile creare il proprio ambiente personalizzato tramite un tool che dispone di alcuni template di stanza alcuni gratuiti altri acquistabili tramite criptovalute. Una volta scelto lo spazio è possibile aggiungere alcuni oggetti forniti da Spatial che però si

riducono a pochi elementi di una libreria. L'asset più personalizzabile è un Frame dentro al quale è possibile aggiungere immagini, video o NFT che si possiedono. Il tutto potendo modificare posizione nello spazio, rotazione e scala.

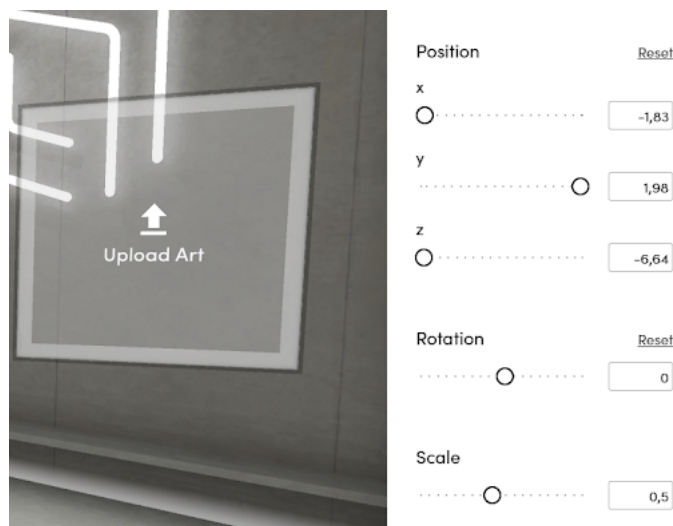


Figura 2.11. Editor quadri digitali 'Spatial'

Si possono creare ambienti totalmente custom sia negli oggetti che costituiscono l'ambiente stesso sia nell'architettura stessa della stanza, tuttavia è necessario pre-modellarli e quindi caricarli sulla piattaforma, questa è certamente una limitazione per molti utenti che si avvicinano a Spatial.

2.5.6 Portali



Figura 2.12. Portali per altre stanze 'Spatial'

Si possono inoltre aggiungere al proprio ambiente dei Portali che collegano la propria stanza ad una qualsiasi altra stanza, sicuramente questa feature è molto interessante siccome rende Spatial e tutte le sue stanze una possibile porta sul metaverso permettendo di rendere i confini tra le stanze e community molto più labili.

Spatial è sicuramente una piattaforma interessante per quel che riguarda il concetto di metaverso siccome cerca di creare spazi virtuali, improntati all'esposizione artistica e museale, permettendo agli utenti di comunicare e di interagire tra loro, tuttavia fallisce siccome privo di una community interessata e coinvolta. Esplorando le diverse stanze si troverà qualcosa di interessante e che ti incentiva ad esplorare meglio l'ambiente tuttavia manca la partecipazione della community e spesso l'esplorazione risulta vuota e fine a sé stessa.

2.6 Virtual Expo

Virtual Expo si tratta di una piattaforma sviluppata e mantenuta internamente da Funix, è stata pensata per realizzare spazi espositivi virtuali in maniera rapida ed intuitiva. Si divide in due parti distinte, una parte dedicata al cliente e quindi a colui che si occuperà dell'inserimento dei contenuti definita Dashboard, ed una seconda parte per la visualizzazione degli stessi contenuti e quindi con possibilità di interazione con l'ambiente dedicata a chi navigherà l'esposizione virtuale. La piattaforma è sviluppata con Unity WebGL saranno quindi necessarie alcune accortezze, soprattutto per l'upload di modelli, che permettano la corretta navigazione dello spazio espositivo virtuale.

2.6.1 Dashboard

Allestimento

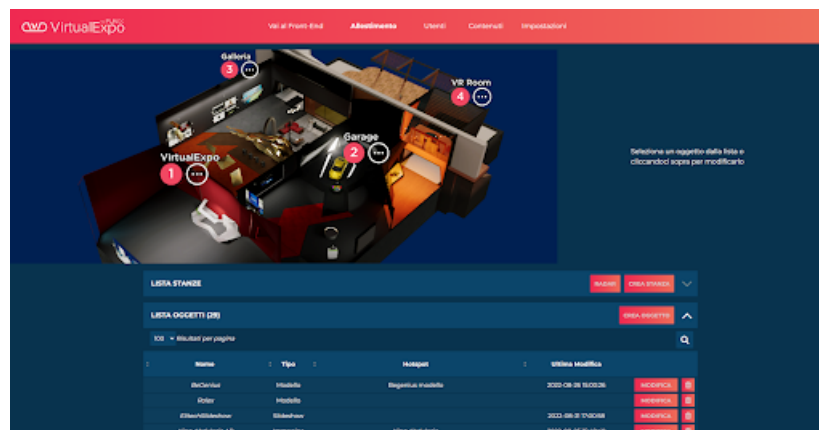


Figura 2.13. UI allestimento 'Virtual Expo'

In questa sezione è possibile gestire gli elementi modificabili (detti Oggetti) all'interno dell'applicazione. Sono disponibili 4 categorie di oggetti: Modello 3D, Elemento 2D, Slide-show e Hotspot. Questi elementi verranno creati e definiti attraverso proprietà specifiche ma condivideranno la possibilità di essere posizionati liberamente all'interno dell'ambiente 3D. In questa sezione si potranno inoltre suddividere i diversi oggetti in categorie così da averli raggruppati. Nella pagina di allestimento si avrà un editor visuale all'interno del quale potranno essere posizionati tutti gli oggetti, i quali una volta posizionati potranno essere manipolati tramite il classico Gizmo di Unity che permette quindi modifica di posizionamento quindi definizione delle coordinate spaziali, rotazione sugli assi X e Y e scala quindi le dimensioni dell'oggetto.

Tutte le informazioni di modifica verranno mostrate in tempo reale nell'Inspector a lato dell'editor visuale, dove dovranno essere salvate le modifiche una volta eseguite. Oltre alla possibilità di definire posizione, rotazione e scala sarà possibile definire degli Hotspot associati agli oggetti ovvero punti cliccabili che definiranno una descrizione dell'oggetto,

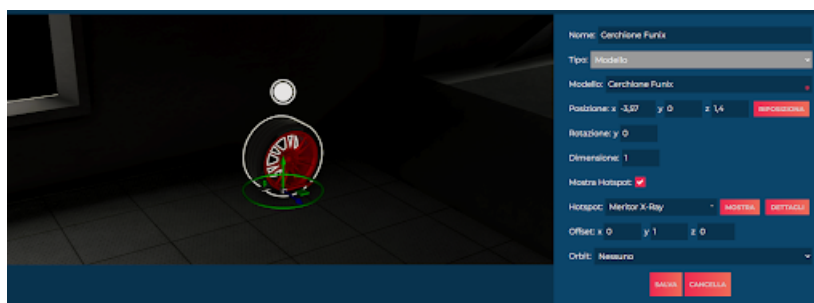


Figura 2.14. Inspector Oggetto 'Virtual Expo'

questa stessa proprietà sarà modificabile per posizione nello spazio ma non per rotazione in quanto sempre rivolto verso l'utente. I diversi oggetti definibili nella sezione allestimento sono quindi:

- Modello: quindi modelli 3D precedentemente caricati sulla piattaforma da Funix e liberamente posizionabili dall'Editor
- Immagine: una singola immagine in formato PNG o JPG, liberamente caricabile dall'editor
- Slideshow: ovvero una collezione di immagini con dimensioni e posizioni a scelta dell'editor suddivise per pagine
- Hotspot: ovvero un'icona interagibile non necessariamente collegata a nessun oggetto (utile per fornire informazioni generiche)
- Modello con Immagine: una combinazione di modello e immagine. Alcuni modelli potranno prevedere di avere un'immagine in una posizione specifica in modo da poter essere posizionati e spostati insieme (una cornice con il suo quadro o una televisione accesa)

Stanze

Come detto precedentemente gli oggetti potranno essere raggruppati in categorie, le quali sono concepite, ma non limitate a questo, come suddivisioni dello spazio per raggruppare i modelli in questa caso verranno definite Stanza, si potrà quindi:

- Specificare un nome per il gruppo di oggetti visibile nella vista dall'alto
- Inserire un Hotspot descrittivo
- Creare un'ancora sul Radar e definire il punto di vista iniziale in caso le Stanze vengano usate per suddividere lo spazio espositivo

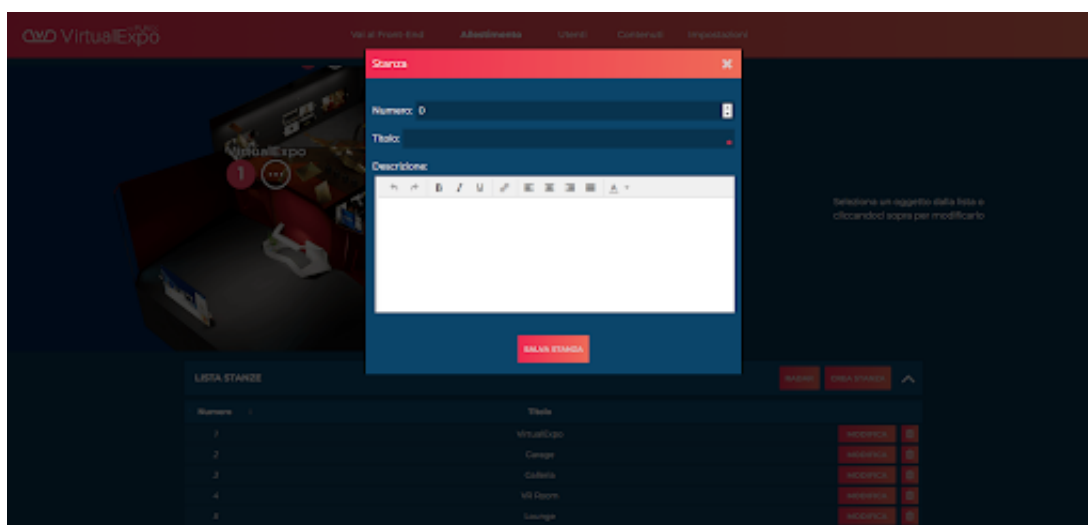


Figura 2.15. Creazione stanza 'Virtual Expo'

Radar

Sul radar sarà possibile definire la posizione delle stanze, il radar stesso sarà utilizzato come strumento di navigazione nel front-end.

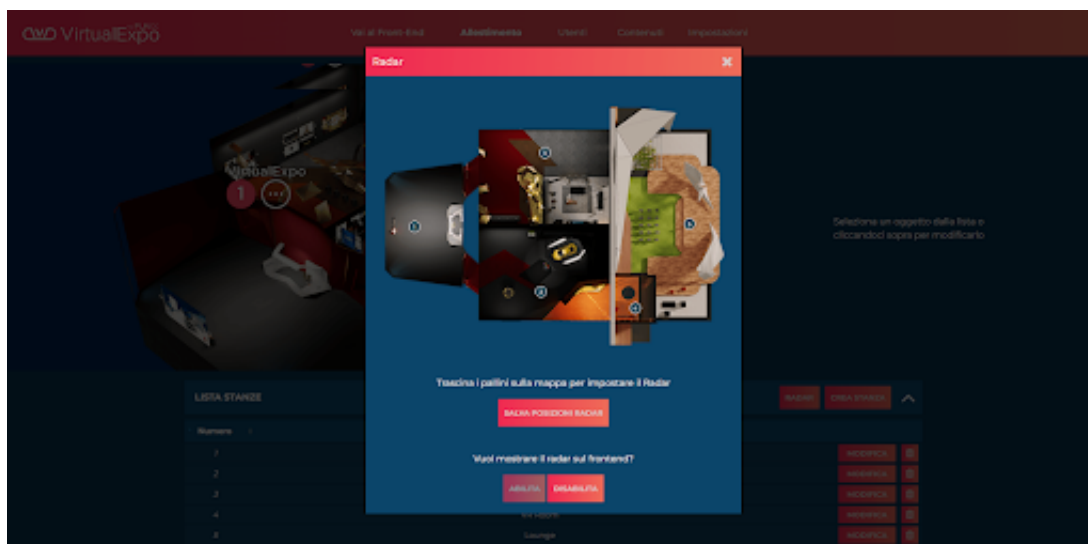


Figura 2.16. Editor Radar 'Virtual Expo'

Modelli

Nella sezione modelli sarà possibile visualizzare tutti i modelli che siano stati caricati sulla piattaforma e gestire i loro dati sul database, come per altri oggetti sarà possibile definire le Categorie per suddividere logicamente i prodotti, suddivisione visibile poi nella sezione di allestimento. Questi modelli potranno essere o meno autoprodotti ma dovranno essere caricati comunque da Funix per questioni tecniche di ottimizzazione.

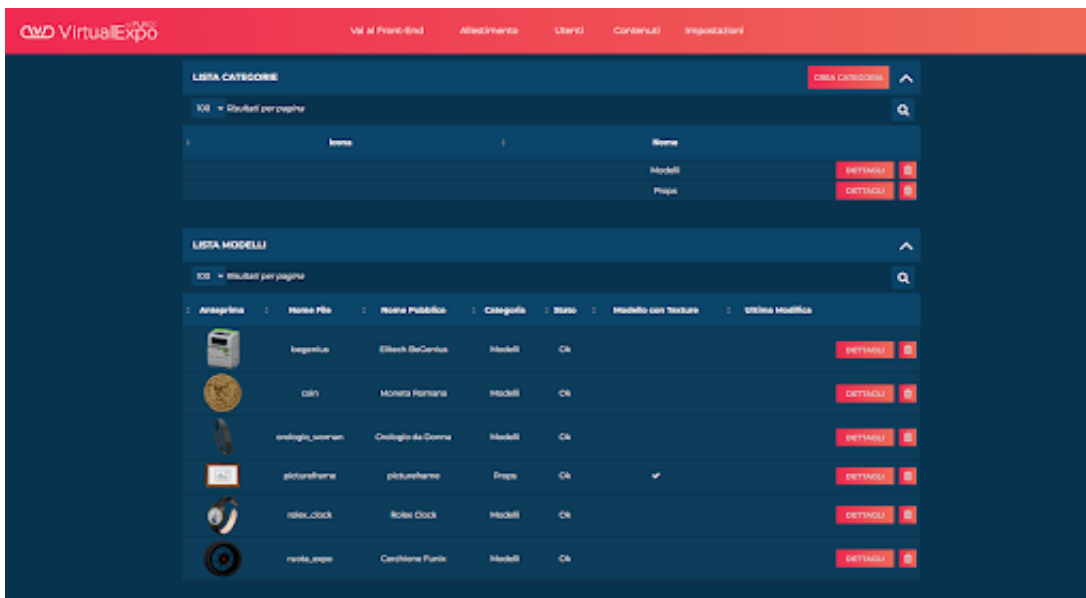


Figura 2.17. Lista dei modelli presenti in 'Virtual Expo'

Per ogni oggetto sarà possibile specificare un nome esterno (o visualizzato) per identificarlo, i modelli nella sezione allestimento potranno essere controllati per capire se sono presenti o meno delle voci di Database che non abbiano un modello fisico corrispondente sul server.

Orbit



Figura 2.18. Visualizzazione Orbit 'Virtual Expo'

Si tratta di una proprietà dedicata ai modelli che permetterà sul front-end una rotazione con closeup sull'oggetto. Sarà di 2 tipi diversi:

- Closeup: che attiverà un turntable sull'oggetto all'interno dell'ambiente stesso
- Limbo: si attiverà un turntable sull'oggetto in un limbo che permetterà di avvicinarsi di più all'oggetto e vederlo in maggior dettaglio

In entrambe le viste, su un pannello laterale verrà mostrato il nome del modello e in caso l'oggetto abbia un Hotspot associato, una breve descrizione.

Hotspot

Come detto precedentemente gli Hotspot sono gli elementi che permettono di inserire contenuti informativi all'interno della piattaforma. Vengono generati in una sezione dedicata e possono essere, o meno, legati ad altre tipologie di oggetti. Gli Hotspot si potranno dividere in categorie, alle quali si potrà associare un'icona personalizzata. Si potranno definire 3 tipologie di Hotspot:

- Audio: che permette di inserire una traccia audio in formato mp3 all'interno di VirtualExpo con proprietà quali Loop, quindi una riproduzione continua dell'audio una volta terminato; o in alternativa si potrà definire la proprietà Audio Spazializzato che riproduce l'audio nel punto in viene posizionato l'hotspot facendo decrescere il volume man mano che ci si allontana dall'hotspot rafforzando l'illusione che appartenga all'oggetto stesso o alla stanza cui l'hotspot è associato.

Figura 2.19. Lista Hotspot 'Virtual Expo'

- **Link:** si tratta di un collegamento ad un'altra pagina Web che si potrà aprire tramite popup (iFrame) o in una nuova scheda del browser.
- **Popup:** all'interazione si aprirà una finestra con contenuti definiti dall'utente quali titolo, sottotitoli, immagini, descrizione e altri, ognuno dei quali opzionali.

Slideshow

Nella sezione degli slideshow è possibile creare delle collezioni di immagini suddivisibili per pagine che l'utente potrà scorrere liberamente. Ogni slideshow è definito da:

- **Titolo:** identifica lo stesso slideshow nella sezione Allestimento.
- **Dimensioni:** vengono definite in pixel quindi larghezza e altezza dello slideshow.
- **Autoplay:** quindi un cambio automatico o meno delle diverse slide, Tempo di autoplay ovvero il tempo di attesa tra un cambio automatico e l'altro delle slide.
- **Colori:** si avrà la possibilità di definire lo sfondo dello slideshow che creerà eventuali bordi alle immagini a seconda delle loro dimensioni, oppure le Immagini quindi il colore che verrà applicato alle eventuali trasparenze delle immagini nella conversione in Jpeg.
- **Pagine** quindi quante slide saranno presenti nel carosello

In ogni pagina dello slideshow sarà possibile aggiungere uno o più elementi con i propri parametri quindi ogni immagine, se presente, avrà definita posizione nelle coordinate X e Y, dimensioni e Hotspot che renderà l'immagine cliccabile.

Statistiche

Nella sezione dashboard oltre alla gestione dei diversi oggetti che popolano il virtual expo, sarà anche possibile gestire le statistiche riguardanti alla gestione degli utenti quindi:

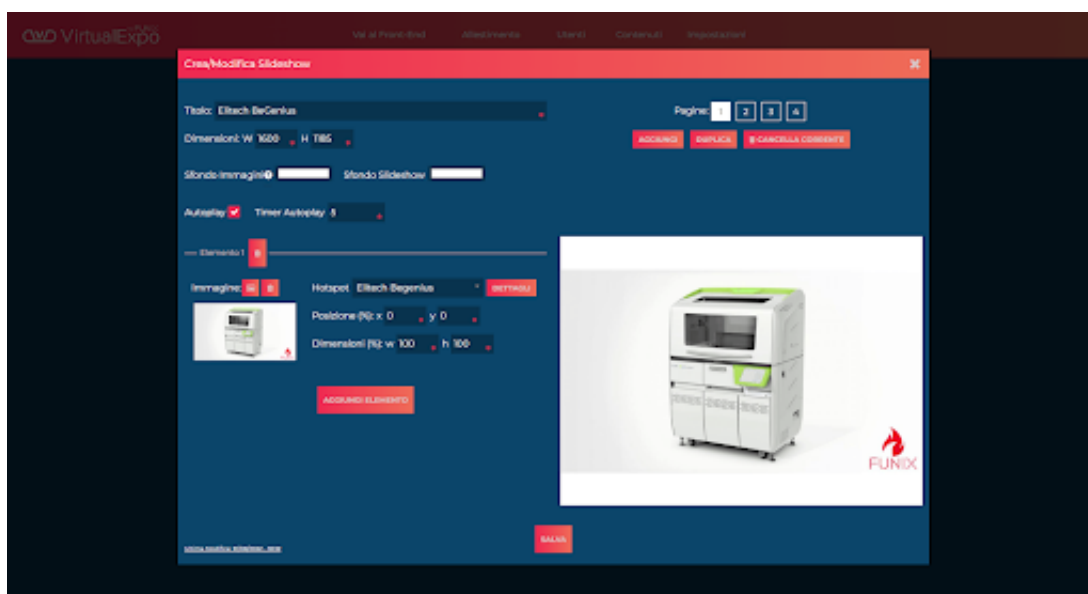


Figura 2.20. Inspector Slideshow 'Virtual Expo'

- Login/logout: connessione e disconnessione degli utenti della piattaforma
- Hotspot aperti: i contenuti attivati dagli utenti
- Editor Log: Dove verrà tenuta traccia la creazione, modifica e cancellazione di tutti gli Oggetti, Hotspot, Modelli e Slideshow. Viene visualizzata la data di ultima modifica e quindi si potrà osservare da un popup tutto il log di un determinato Oggetto.

Le statistiche si possono esportare in Excel ed esiste la possibilità di escludere dalle metriche gli editori.

Utenti

Modulo opzionale della piattaforma, siccome la stessa non prevede necessariamente un login degli utenti registrati, la sua esclusione o meno prevederà delle differenze lato Front-end. Sono previste due tipologie di utenti:

- Utenti Base: se attivo il modulo utenti, saranno gli utenti che si potranno registrare da front-end ed avere accesso alla sola parte di VirtualExpo
- Editori: sono gli utenti con accesso ad entrambe le parti di VirtualExpo avendo possibilità di accedere a tutte le funzionalità.

Ogni utente è definito per: Email, Password, Username, Nome e Cognome. Con possibilità di integrare informazioni ulteriori.

2.6.2 Front-End

Virtual Expo



Figura 2.21. Vista della pagina di 'Virtual Expo'

Il front-end si presenta come un'area espositiva suddivisibile in ambienti che permette di accedere a tutti i contenuti posizionati tramite dashboard. L'accesso può essere subordinato ad una registrazione o meno. I contenuti saranno sempre liberamente fruibili, gli hotspot appariranno come icone, diverse a seconda della loro categoria ed apriranno informazioni collegate al contenuto scelto, quando aperti impediranno la navigazione.

Navigazione

L'applicazione si apre su una vista aerea dalla quale si può accedere agli ambienti del VirtualExpo, quindi dalla vista aerea si può passare alla vista interna degli ambienti tramite un click sul modello 3D, o sul radar in caso siano state fissate delle ancore (in allestimento). Nella vista esterna si potrà ruotare liberamente l'ambientazione così da osservare esternamente tutta la planimetria da ogni angolazione desiderata. Una volta all'interno si distinguerà una navigazione tra prima e terza persona, e tra puntamento tramite mouse o tramite frecce direzionali. Tutte queste possibilità di movimento convivono nello stesso ambiente e sta solamente all'utente decidere quali di queste utilizzare.



Figura 2.22. Navigazione in 'Virtual Expo'

Menu Laterale



Figura 2.23. Menu laterale in 'Virtual Expo'

All'interno dell'applicazione è possibile avere un menu laterale parzialmente configurabile, dove è possibile gestire la presenza o meno un barra di ricerca degli Hotspot, un sistema di preferiti, link a pagine esterne e l'accesso nel caso fosse implementato il modulo utenti, profilo utente in cui gestire le proprie informazione e quindi tasto logout.

Integrazioni Custom

VirtualExpo essendo una piattaforma interna e totalmente realizzata da Funix, prevede la possibilità di realizzare feature custom (animazioni, minigiochi . . .) su richiesta, proprio perché lo scopo del progetto è quello di realizzare un modello sul quale lavorare per ogni esigenza futura dei clienti.

Caratteristiche Tecniche

La Dashboard è sviluppata in Php+MySQL con front-end in HTML e Javascript. Non sono integrati CMS terzi e vengono integrate alcune librerie consolidate e facilmente mantenibili (es. jQuery). Necessita di un server LAMP (Linux Apache MySql Php) e si chiederà la condivisione degli accessi al Database MySQL per le integrazioni specifiche del cliente. La Dashboard, pur essendo responsive, non è pensata per essere mobile ready in quanto la parte di Allestimento è di difficile se non impossibile gestione in un contesto mobile. Il server su cui verrà ospitata la piattaforma non è fornito con l'applicazione. La parte in WebGL è sviluppata con Unity3d con interfaccia in HTML e Javascript. Le applicazioni in Unity WebGL sono compatibili con i dispositivi mobile. Tenendo conto delle limitazioni del contesto mobile, le sezioni interattive sono state realizzate per una fruizione quanto più possibile simile a quella del sito in contesto desktop. Le applicazioni sono compatibili con i più moderni browser che fanno uso della tecnologia WebGL, ed ottimizzate per migliorare i tempi di caricamento in base al contenuto scelto. Per il caricamento dei contenuti, sarà necessario l'accesso FTP al server dove sarà ospitata la piattaforma. Tutti i modelli 3d dovranno essere ottimizzati ed integrati da Funix. Dopo averli integrati nella piattaforma diventeranno selezionabili tramite il menù "Modello" per gli elementi modificabili per creare l'allestimento. Tutti gli altri contenuti possono essere caricati liberamente dall'utente senza limiti e non prevedono l'intervento di Funix.

Capitolo 3

Metodo

3.1 Task Principali

Il mio lavoro durante il tirocinio al progetto di VirtualExpo si è occupato principalmente di:

- Avatar: quindi tutto ciò che riguarda il movimento, l'animazione, il cambio di vista tra prima e terza persona e la possibilità di scaricare un avatar personalizzato.
- Multiutente: ovvero la possibilità di avere più utenti compartecipanti ad una stanza dovendo gestire la coerenza dei movimenti spaziali per tutti gli utenti, la corretta trasmissione di informazioni e l'entrata o uscita di un nuovo partecipante.
- Comunicazione: tutto ciò che riguarda la comunicazione tra utenti dalla chat di testo, alla trasmissione dei canali audio e video tramite webcam e la condivisione del proprio video all'interno di una scena in Unity

Nel corso del capitolo verranno quindi sviscerate le tecnologie, il funzionamento delle parti e le soluzioni trovate e applicate per risolvere problematiche durante il periodo di partecipazione al tirocinio.

3.2 Avatar

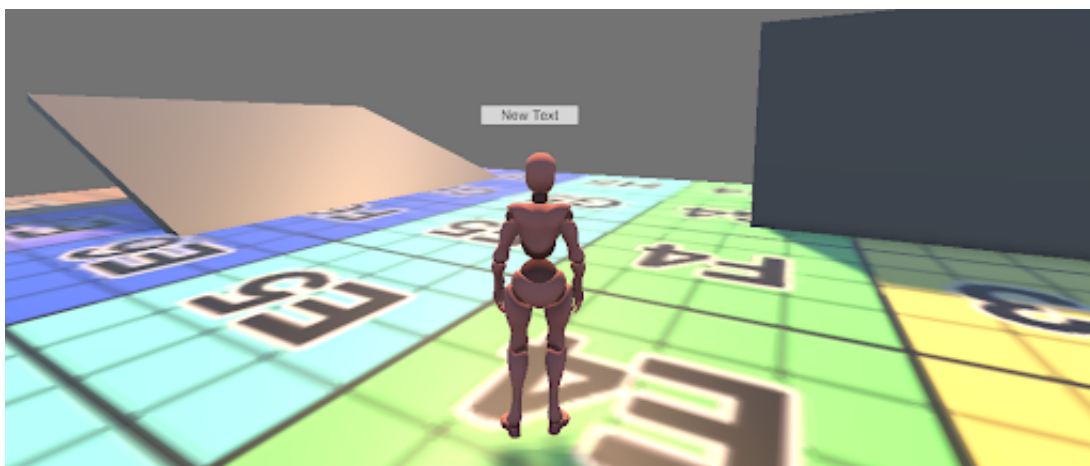


Figura 3.1. Avatar nella sezione di Test

All'interno della piattaforma Virtual Expo come visto sarà possibile muoversi tramite un avatar, con possibilità di cambiare la vista tra prima e terza persona, movimento tramite input da tastiera e click del mouse è stata inoltre sviluppata la possibilità di cambiare il proprio avatar con uno generato tramite ReadyPlayerMe :

- Prima persona: la vista dell'utente è in prima persona con possibilità di zoomare
- Terza Persona: la vista dell'utente è in terza persona mostrando quindi l'avatar dell'utente a figura intera
- Punta e clicca: l'utente si può muovere nell'ambiente tramite click del mouse
- Tasti direzionali: l'utente di può muovere nell'ambiente tramite frecce direzionali o i tasti WASD
- Cambio modello avatar

3.2.1 Prima e Terza Persona

Definiamo quindi il passaggio di punti di vista tra prima e terza persona. Per Funix era necessario avere la possibilità di scegliere il punto di vista così da permettere ad ogni utente di scegliere liberamente la prospettiva che preferisce, rendendo quindi la navigazione all'interno dei virtual tour personale e diversificata secondo le esigenze. Si è quindi deciso di scrivere uno script che rendesse possibile il passaggio tra prima e terza persona in maniera intuitiva e leggera dal punto di vista computazionale, tramite una soluzione all in one. Per fare questo si lavora direttamente sulla posizione della camera e sul FOV (Field of View) della stessa per creare un movimento coerente, unico e senza tagli. Per passare

dal punto di vista prima a terza persona basterà da computer ruotare la Mouse Scroll wheel mentre da mobile tramite la gesture del pinch.

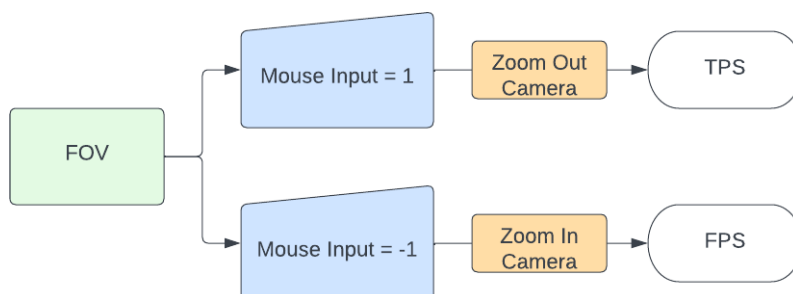


Figura 3.2. Diagramma script cambio POV

Lo script che gestisce questo passaggio non è complesso, infatti è capace di percepire gli input forniti dal mouse e in base al senso di rotazione della rotella, fornire uno zoom in e zoom out della camera fino a determinate posizioni decise a priori. Una volta raggiunte, o superate, certe soglie verrà impostato il raggiungimento delle stesse e quindi si attiveranno delle funzioni che definiranno alcuni parametri che non solo fissano il punto di vista ma che modificano anche il movimento dell'avatar all'interno della scena. Importante feature per rendere più pulita la transizione da un punto di vista all'altro è quella dell'aggiunta di uno shader di dithering alle texture che costituiscono l'avatar, questo shader all'avvicinarsi della camera al modello dell'avatar verrà istanziato e modificato rendendolo meno o più presente, nel momento in cui si attiva la mesh diventerà semitrasparente.

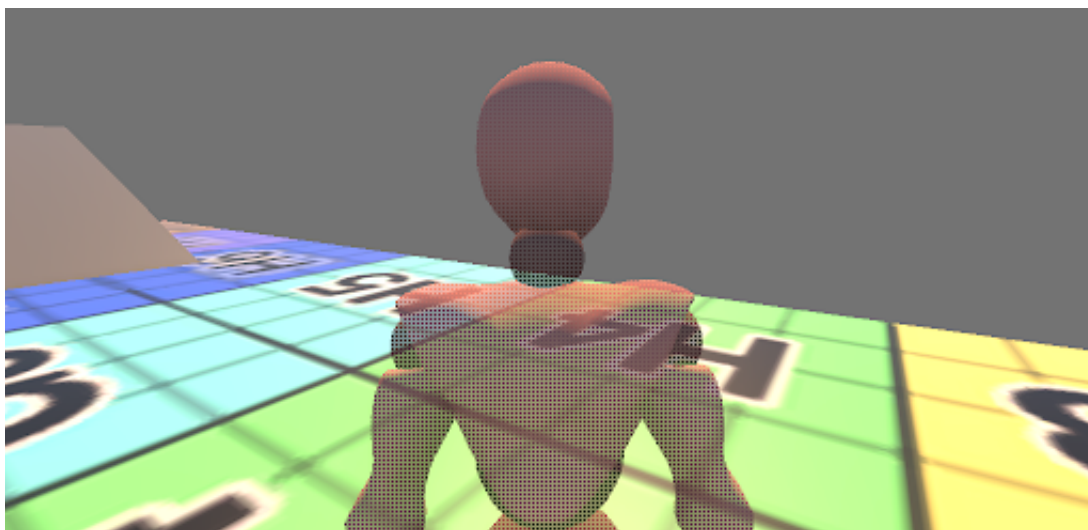


Figura 3.3. Effetto di dither sull'avatar

Per gestire questa funzionalità verrà utilizzato uno script legato al modello chiamato 'Disappear By Distance'. Questo script non appena viene istanziato nella fase di `Start()`, scandaglia tutti i componenti figli e ne individua le `SkinnedMeshRenderers` ai quali viene sostituito lo shader con lo shader di Dither.

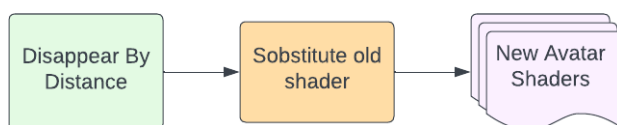


Figura 3.4. Diagramma script cambio shader

Nella fase di `LateUpdate()` viene calcolata la distanza dall'avatar e in base a questa si definirà un valore detto `disappearingValue` il quale viene calcolato in base a `distanceFromPlayer / disappearingDistance` il cui risultato verrà proiettato su una curva.

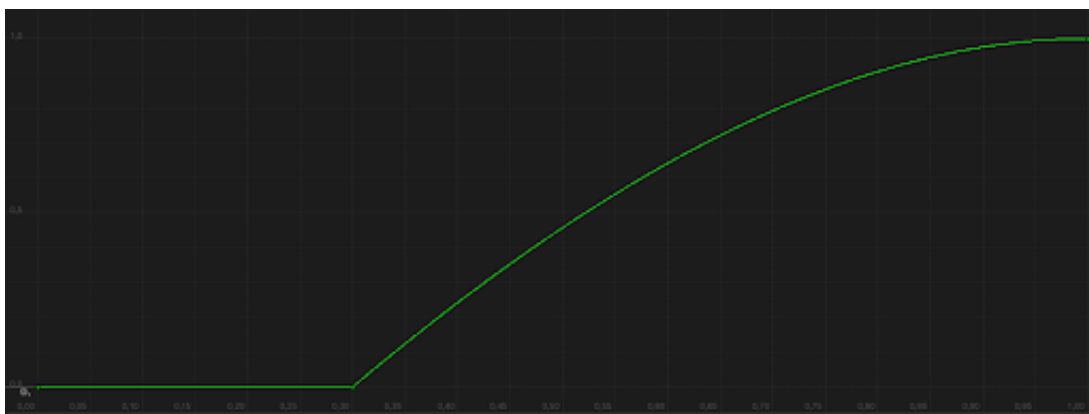


Figura 3.5. Curva applicata allo shader

Quindi si leggono tutti i material con shader integrato che costituiscono l'avatar, e verrà modificato il valore di `opacity` che agisce sull'effetto di Dither, ottenendo così una transizione visivamente più gradevole e meno sporca. All'interno dello script che controlla il movimento dell'avatar è posta una funzione chiamata `NewModelLoaded()`, questa viene chiamata non appena l'utente decide di cambiare il modello del proprio avatar tramite l'estensione di Ready Player Me, sarà necessaria invocarla per ottenere gli shader dei nuovi materiali così da poterli modificare successivamente e quindi ottenere l'effetto di Dither anche con il nuovo modello.

L'avatar avrà associata a sé una camera che permette all'utente di visualizzare la scena, questa camera tuttavia non si tratta della semplice camera di Unity, bensì fa parte del package di Unity chiamato Cinemachine [UnityCam]. Si è scelto di utilizzare questo potente tool con diverse funzionalità principalmente per sfruttare la funzionalità avanzata denominata Cinemachine Collider, questo perchè altrimenti con la semplice camera

di Unity non si sarebbero rilevate le collisioni facendo così compenetrare la stessa nelle pareti, distruggendo in maniera decisa il senso di immersione. Per utilizzare il tool quindi serve creare una `Virtual Camera` ovvero un `GameObject` con diversi parametri. Tra i parametri presenti uno dei più importanti è `Follow`, che definisce il `gameobject` da seguire permettendo alla virtual camera di posizionare la camera di Unity in quella posizione anche se si muove, nel nostro caso si tratterà di una seconda camera in unity ma disabilitata questo perchè ci è utile per estrarre alcuni parametri non manipolabili nella `Cinemachine Camera`, creando una camera proxy invece ci è possibile intervenire sulla `Cinemachine Camera` più liberamente. Altro importante parametro è quello di `Look At` che definisce il `GameObject` specifico a cui puntare facendo ruotare automaticamente la camera per far sì che sia sempre rivolta a quello specifico `Gameobject`, nel nostro caso si tratterà di un `empty` posizionato sulla testa dell'avatar. Altro componente da configurare è il `Cinemachine Brain` responsabile di monitorare tutte le attività della `Virtual Camera`, sono presenti diverse proprietà che ne caratterizzano l'aggiornamento della posizione e della rotazione o la curva a cui la camera tenderà muovendosi. Ultimo importante componente che rende la camera dell'avatar efficace è il `Cinemachine Collider`, un'estensione che post processa la posizione finale della virtual camera, cercando di preservare la linea visiva con il `Look At` target della virtual camera e lo fa spingendo la `Virtual Camera` dai `GameObject` che ostruiscono la vista. Funziona tramite un `Physic Raycaster` che individua oggetti tridimensionali con un collider così da poter muovere e spostare la `Virtual camera` di conseguenza.

3.2.2 Movimento Avatar

Il movimento dell'avatar come detto precedentemente viene suddiviso tra movimento tramite click o tramite frecce direzionali e WASD, questa scelta rende l'utente da PC libero di muoversi nel modo che preferisce e permette all'utente mobile di muoversi senza bisogno di alcuna periferica esterna bastando il click direttamente sullo schermo. Quindi per il movimento dell'avatar all'interno della scena viene utilizzato il componente chiamato `NavMeshAgent` che permette ad un qualsiasi oggetto a cui viene collegato il componente di muoversi nella scena tramite `NavMesh` ovvero lo spazio calpestabile calcolato da Unity.

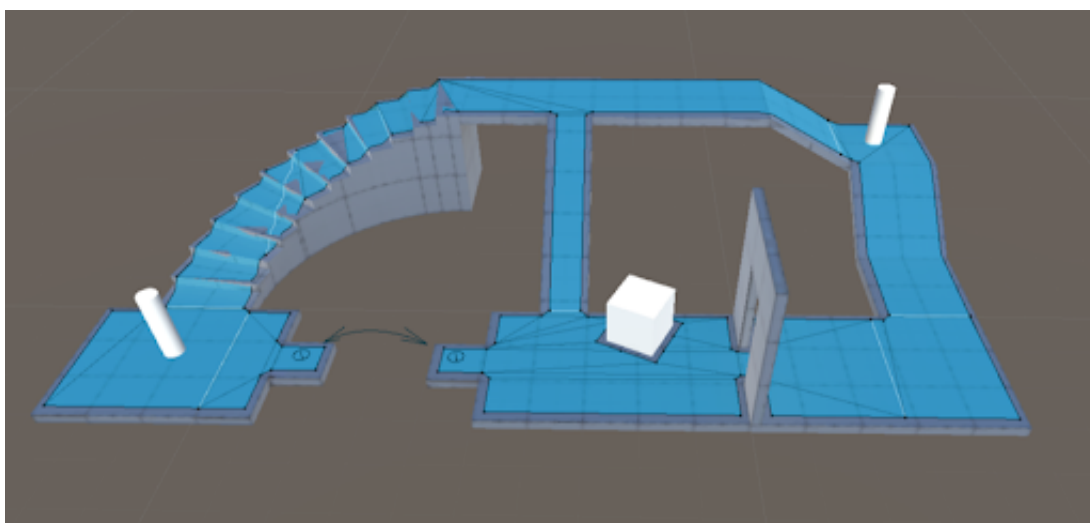


Figura 3.6. Navmesh su cui si muovono i NavMeshAgent

Il sistema di navigazione tramite NavMesh [UnityNavmesh] permette quindi ad un oggetto qualsiasi di muoversi nella scena, l'area calpestabile viene automaticamente estratta dalla geometria dell'ambiente. Si possono avere anche ostacoli dinamici che alterano la navigazione del character a runtime. Il componente di NavMeshAgent ha diversi modificabili tuttavia nessuno di questi è degno di nota per la trattazione del funzionamento.

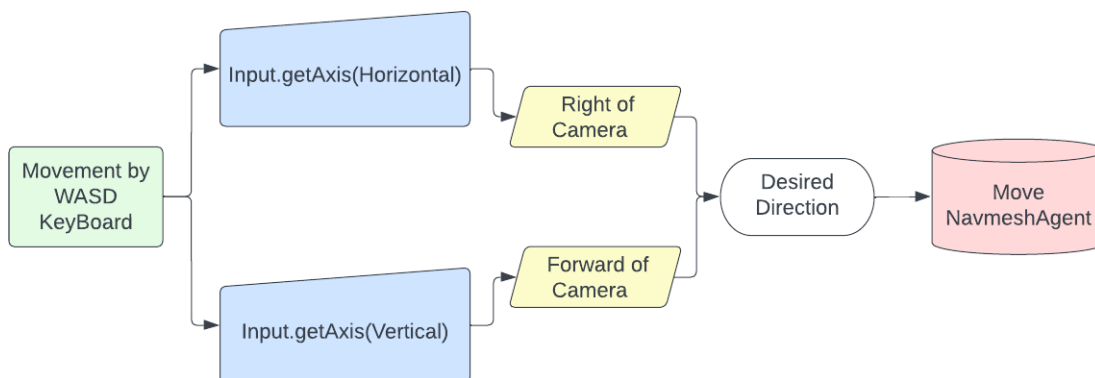


Figura 3.7. Diagramma Script Movimento tramite input direzionali

Lo script che permette la navigazione tramite input direzionali è capace di estrarre le informazioni di input orizzontali e verticali tramite `Input.GetAxis()`. I valori ottenuti verranno quindi combinati con i valori di `Right` e `Forward` della camera in Unity, reciprocamente asse Rosso e Blu della trasformata della camera, questi valori ci permettono di muovere l'avatar coerentemente rispetto alla camera. Quindi ottenuti questi 2 `Vector3` che definiscono il punto di riferimento verso cui muoversi, rispetto alla camera, verranno

combinati con i valori di Input direzionali ottenendo il Vector3 "desiredMoveDirection" della direzione desiderata, il quale viene successivamente normalizzato per evitare di avere una velocità di movimento raddoppiata quando ci si sposta in obliquo. Il vettore ottenuto viene moltiplicato per il tempo e per la velocità di camminata preimpostata nell'inspector e quindi applicato al NavMeshAgent tramite la funzione integrata .Move() che applica un movimento relativo alla posizione corrente, quindi si usa la funzione destination che applica al NavMeshAgent la destinazione nel World Space. Nello stesso script che applica il vettore di movimento, viene anche applicata la rotazione al character. Per ottenere la rotazione si estrae il quaternion toRotation che definisce la rotazione verso cui ruotare l'avatar, il quaternion si estrae tramite la funzione LookRotation che crea una rotazione rispetto ai valori di Forward e Upward forniti, ovvero il parametro calcolato desiredMoveDirection. Una volta ottenuto il quaternion desiderato si modificherà la rotazione locale dell'avatar tramite la funzione RotateTowards che richiede una rotazione di partenza, la localRotation dell'avatar, e la rotazione da raggiungere, toRotation, il tutto secondo una velocità calcolata.

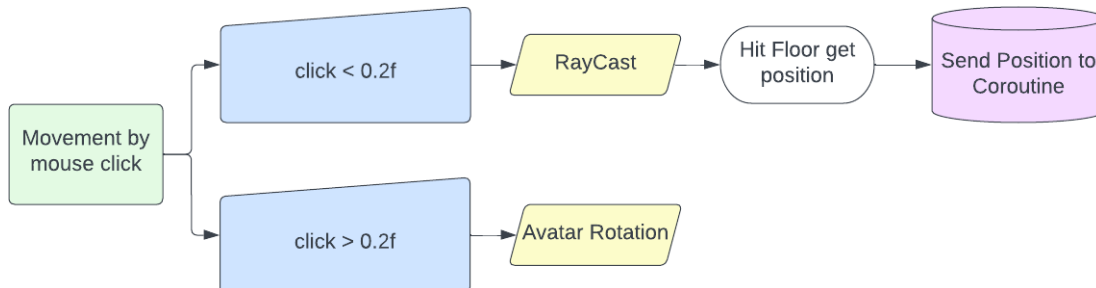


Figura 3.8. Diagramma Script Movimento tramite input click del mouse

Lo script che si occupa del movimento tramite click del mouse ha diversi controlli sul click stesso, `Input.GetMouseButton(0)`, questo perché se il click è minore di 0.2f allora viene interpretato come input di movimento diversamente vorrà dire che si starà ruotando l'avatar, tuttavia ora ci concentreremo sul movimento. Dopo aver verificato che il click è per muoversi si verificherà se aver cliccato un luogo compatibile con il movimento, ovvero la NavMesh in tal caso verrà fatta partire una Coroutine, ovvero funzioni che si interromperanno una volta soddisfatti dei valori. Non appena la coroutine parte verranno impostati alcuni parametri di rotazione relativi alle rotazioni iniziali di camera e avatar, viene definito il quaternion del punto di arrivo con il parametro 'targetRotation', calcolato tramite la posizione di arrivo e posizione finale. Quindi si verifica che il punto di arrivo sia vicino o meno alla posizione di partenza questo perché, come si vedrà successivamente, se il click è sufficientemente vicino non verrà applicata rotazione alla camera così come con l'avatar per evitare un effetto visivo fastidioso e poco elegante.

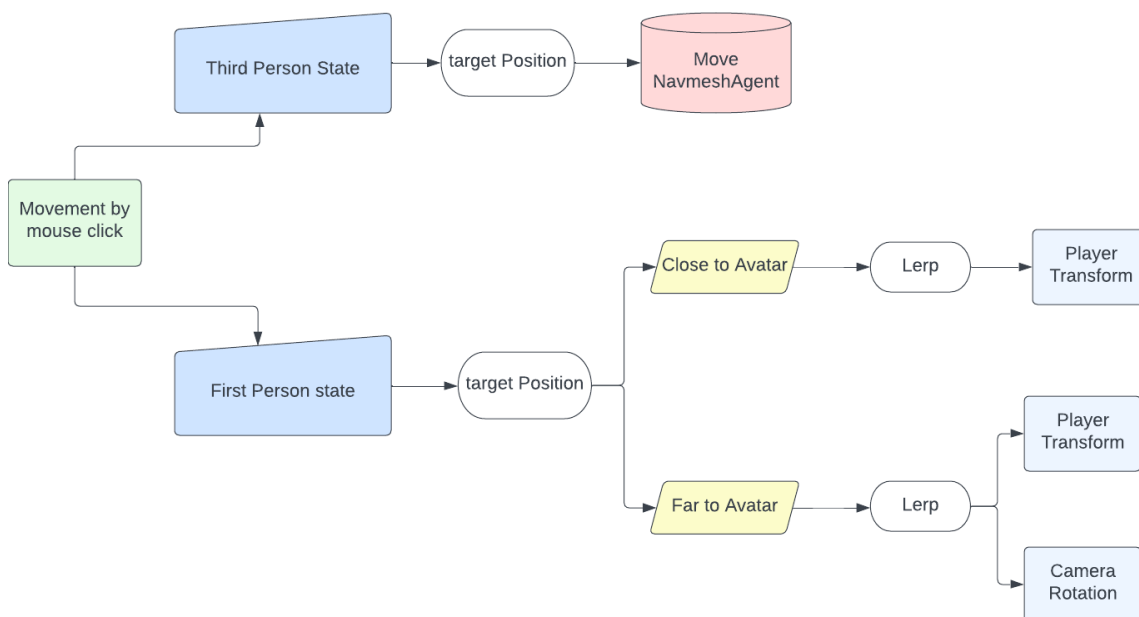


Figura 3.9. Diagramma Script coroutine

Dopo aver verificato che il target scelto sia abbastanza vicino il movimento cambia in base al punto di vista in cui ci troveremo, se siamo in terza persona, si invocherà la funzione `SetAvatarAgentPosition(targetPosition)` con parametro la posizione cliccata, questa farà muovere l'avatar tramite la funzione del NavMeshAgent `SetDestinatio()`. Se invece siamo in prima persona verrà fatta distinzione sul fatto che il target sia vicino o meno, nel caso lo fosse l'avatar verrà mosso tramite una `Vector3.Lerp`, una funzione che interpola i valori da un punto A ad un punto B secondo un parametro 't' che può crescere secondo diverse curve così da dare la sensazione di accelerazione. Se quindi il target point è vicino non viene applicata alcuna rotazione alla camera così da non avere effetti sgradevoli. Se invece il target è sufficientemente lontano allora si userà la stessa funzione di Lerp applicata all'avatar nel caso precedente la quale verrà anche applicata alla camera. Si è scelto di usare le funzioni di interpolazione `Lerp` per muovere il character in prima persona perchè nonostante non seguano la `NavMesh`, facendo proseguire in linea retta l'avatar, restituiscono un effetto visivo e un'usabilità migliore rispetto al `NavMeshAgent` questo perchè il secondo muove sì l'avatar correttamente sulla navmesh ma non applica nessuna modifica alla camera siccome il target point sicuramente è nel cono visivo e l'effetto finale sarebbe decisamente sporco.

3.2.3 Cambio Avatar - ReadyPlayerMe

Ready Player Me [RPMdoc] si tratta di un piattaforma per avatar personalizzati compatibile in diversi game engine e quindi applicazioni. Viene infatti supportato nei Web Browser, Unity, Unreal, React Native, Android e iOS Native. Tramite Ready Player Me

si invitano gli user a creare avatar personalizzati e ad usarli all'interno delle applicazioni, il sistema ha due componenti:

- Gli user possono creare avatar usando l'Avatar Creator dentro alle applicazioni, sul proprio sito, o tramite il proprio avatar di Ready Player Me
- Integrazione della SDK di Ready Player Me all'interno dell'applicazione

Il sistema di Ready Player Me è estremamente efficace siccome ci permette di non impiegare troppe risorse nel creare design, asset e modelli per i propri avatar, inoltre permette di avere avatar completamente personalizzati all'interno degli ambienti virtuali trattandosi di una soluzione cross platform potendo essere implementata in diversi ambienti di sviluppo.



Figura 3.10. Editor avatar Ready Player Me

L'Avatar Creator si tratta di un'interfaccia Web-Based dove gli utenti possono interagire e creare i propri avatar si può implementare ed utilizzare con diverse soluzioni:

- Gli utenti si possono registrare, e poi creare e gestire la propria collezione di avatar dal sito
- Si può implementare la demo dell'avatar creator
- Come utente registrato con partnership di sviluppo si può ottenere l'accesso ad un Avatar Creator personalizzato che consente molteplici opzioni di personalizzazione in base alle richieste.
- Si può integrare l'avatar creator all'interno della propria applicazione

Una volta creati gli avatar questi saranno conservati all'interno dei server di Ready Player Me e potranno essere richiesti dagli utenti e dalle applicazioni come file .glb tramite una URL univoco e persistente.

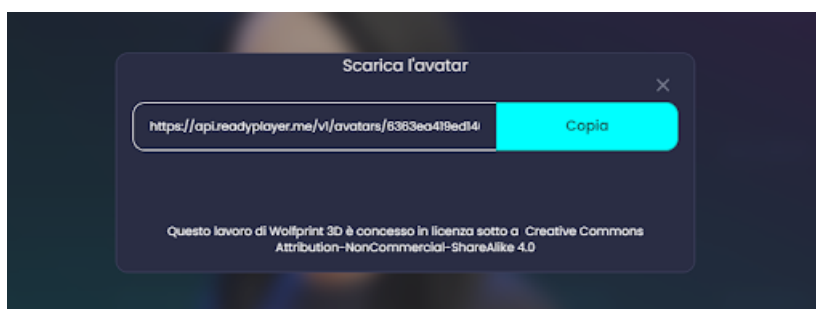


Figura 3.11. Esempio URL univoco fornito

L'Avatar Creator permette quindi agli utenti di creare ed editare i propri avatar, si possono scegliere diverse configurazioni ovvero avatar a figura intera (Full Body) quindi costituiti da un modello 3D completo oppure avatar a Mezzo Busto (Half Body) ovvero avatar di cui si vede soltanto la testa e le mani, utili per applicazioni VR. I dati sia di avatar Full e Half-Body saranno salvati in un unico asset GLB, il quale conterrà tutto ciò che è necessario per renderizzare l'avatar: meshes, textures, blend shapes, skeletal rig e PBR materials. Quindi i dati degli avatar saranno contenuti in un URL univoco il quale ci restituirà un file .glb che può essere recuperato o scaricato. La SDK di Ready Player Me, ovvero un Software Development Kit che indica un insieme di strumenti per lo sviluppo e la documentazione di software, come detto precedentemente può essere integrata in diversi ambienti di sviluppo tuttavia osserveremo soltanto Unity avendolo implementato per il progetto di tesi. L'implementazione è tutto sommato semplice basterà infatti scaricare la versione desiderata ed importarla in Unity nella sezione Custom Package, una volta aggiunto il pacchetto se non ci sono errori di compilazione si potrà osservare la presenza della finestra di Ready Player Me nella toolbar.

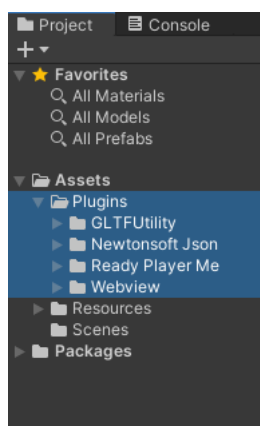


Figura 3.12. Pacchetti plugin RPM in Unity

Nel plugin aggiunto sarà presente `GLTFUtility` che permette di importare ed esportare

glTF files a runtime nell'editor. `Newtonsoft JSON` fornisce i metodi per convertire i `.NET` type con i `JSON`. `ReadyPlayerMe` contiene le funzionalità principali. `Webview` contiene gli overlay per la view su Unity.

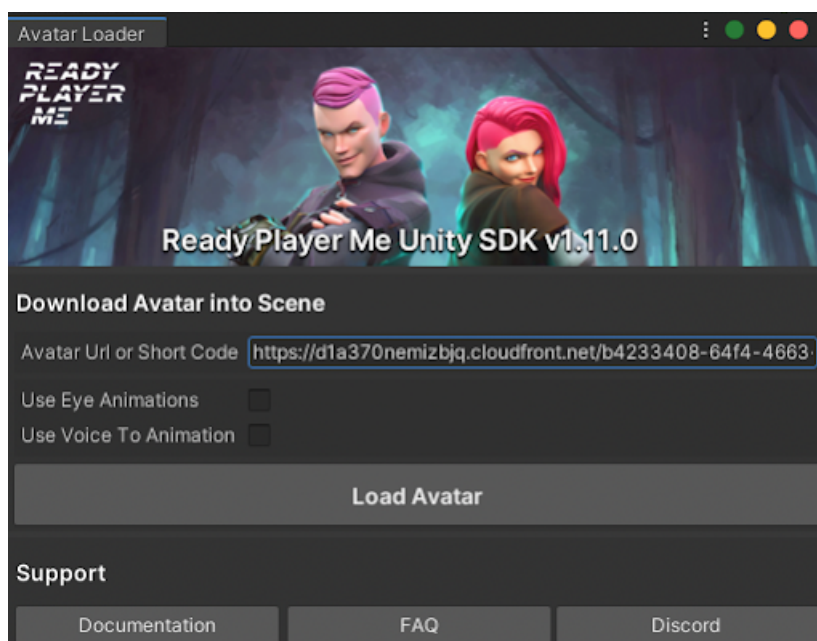


Figura 3.13. Tab RPM in Unity

Una volta integrata la SDK sarà possibile ottenere un avatar tramite la `Loader Window`, nella quale basterà, aggiungere l'URL univoco o lo short code, si possono flaggare altre opzioni per alcune animazioni, che tuttavia non verranno utilizzate nell'implementazione del progetto, quindi una volta caricato avremo il Prefab in scena che se non collegato ad alcun script avvierà le animazioni di default. La `Loader Window` è stata utilizzata nella scena di Unity per testare e debuggare le funzionalità di cambio avatar tramite alcune funzioni ad hoc, questo perché nella scena in WebGL verrà utilizzata una versione che sfrutta le funzionalità del plugin lavorando però dalla `WebView`, ne discuteremo meglio tra poco. Importante caratteristica degli avatar ottenuti tramite Ready Player Me è la possibilità di aggiungere animazioni dal tool online di Mixamo, consentendo di aver un asset di animazioni composito e già strutturato, infatti il modello full body avrà uno scheletro di rig compatibile con le animazioni esportabili. Una volta esportate le animazioni desiderate per importarle basterà trascinare i file nella cartella dedicata, e verranno visualizzate dall'animatore. Per sfruttarle quindi si inseriranno le animazioni nell'Animatore relative agli avatar importati.

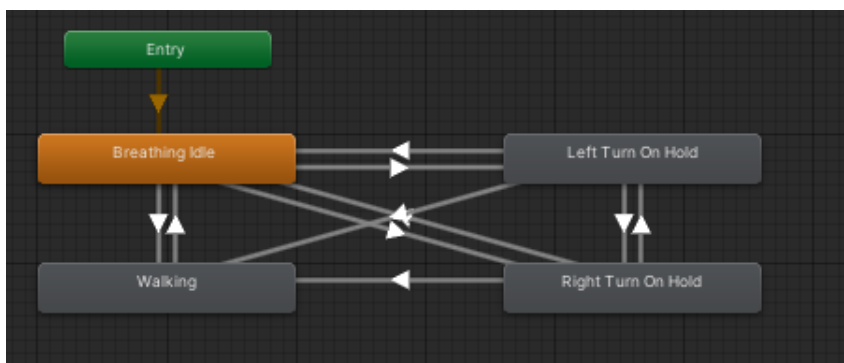


Figura 3.14. Animator Avatar

L'avatar avrà 4 possibili animazioni, una di Idle per quando è fermo e non viene svolta alcuna azione infatti le condizioni che conducono in questo stato saranno velocità di movimento nulla o rotazione del player nulla. Le altre animazioni sono di rotazione sul posto a destra o sinistra e si attivano nel momento in cui il character è fermo e tramite click del mouse si ruota. Infine l'animazione di camminata si attiva nel momento in cui il parametro di velocità del character cambia in base alla formula:

```
//Speed Formula  
currentSpeed = Vector3.Distance(lastPos, playerAvatar.transform.position) / Time.deltaTime;  
lastPos = playerAvatar.transform.position;
```

Figura 3.15. Formula calcolo velocità

Una volta entrati nella pagina relativa alla build in WebGL tra le diverse possibilità di interazione che verranno osservate nella sezione dedicata avremo un pulsante che ci permetterà di caricare ed aprire l'i-frame di Ready Player Me. Una volta cliccato si aprirà a tutto schermo l'i-frame nel quale verrà richiesto il login, se si entra con il proprio account sarà possibile scegliere tra i propri avatar o crearne di nuovi, altrimenti si potrà entrare come utente non registrato e scegliere di crearne uno indipendente. Una volta completato il codice in JavaScript riferito all'i-frame svolgerà tutte le funzioni di creazione del JSON contenente le informazioni dell'avatar inoltre verrà inoltrato un messaggio ad Unity così da triggerare le funzioni necessarie all'istanziamento dell'avatar in scena.

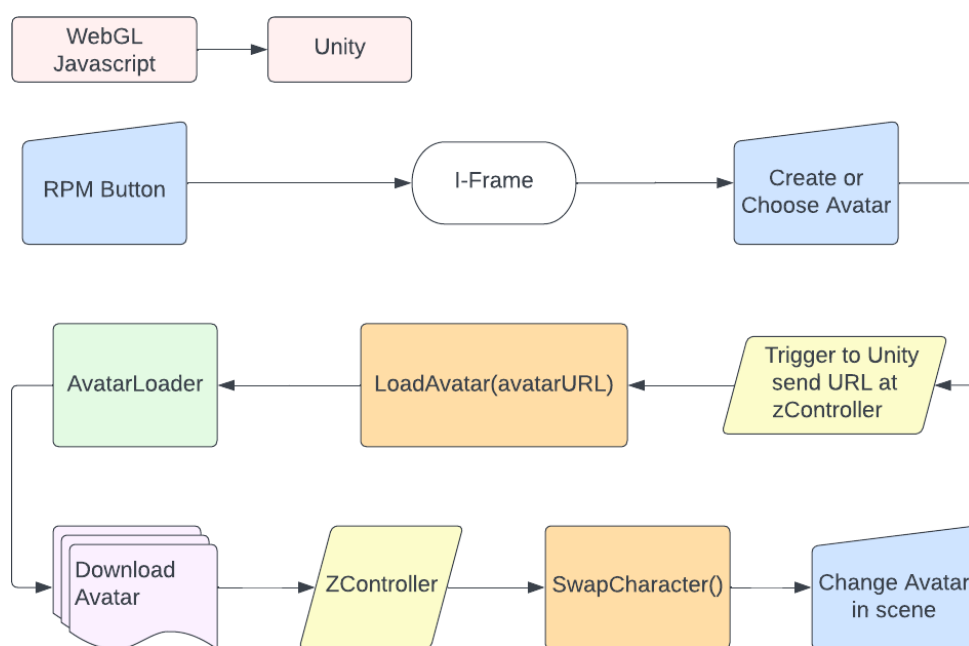


Figura 3.16. Diagramma delle processo di download ed istanziazione di un Avatar da RPM

Come si vede nel diagramma di sopra si può osservare tutto i passaggi che permettono all'avatar di essere istanziato in scena. Una volta personalizzato e confermato l'avatar, lo `zController` in Unity, ovvero un `GameObject` dentro al quale vengono svolte tutte le più importanti funzioni di comunicazione tra JS e Unity, si attiverà e in questo caso specifico verrà invocata la funzione `OnWebViewAvatarGenerated` questa sarà contenuta all'interno di uno script chiamato `WebAvatarLoader`. Quindi una volta chiamato lo script di Unity questo chiamerà tutte le risorse necessarie per stabilire una comunicazione non solo con gli altri componenti di Unity ma anche con i componenti del plugin necessari per svolgere correttamente il download del modello da interfaccia web di ReadyPlayerMe. Una volta ottenuti i permessi, non appena può lo script (fase di `Awake`) verrà invocata la funzione chiamata da Javascript con l'URL univoco del modello e quindi la funzione di `LoadAvatar` che permette il download. Viene quindi invocata una classe del SDK di ReadyPlayerMe chiamata `AvatarLoader` questa grazie all'URL è in grado di scaricare il modello e di istanziarlo in scena associandolo all'Id personale dell'utente il quale è presente in un dizionario, viene quindi chiamata la funzione `SwapCharacter` presente nello `zController`. Una volta istanziato il modello lo sostituirà all'avatar di default, applicando al modello di ReadyPlayerMe l'animatore creato, inoltre sarà in grado di localizzare il modello nello spazio quindi nel luogo in cui l'utente si trova permettendo il cambio dell'avatar in qualsiasi momento. Se si tratta del primo download del modello, quindi con `UserID` vuoto, sarà chiamata la funzione `NewModelLoaded` che modificherà tutte le texture ed i materiali del nuovo modello così da poter avere l'effetto di `Dithering`. Inoltre viene inoltrato un evento al Javascript così che tutti gli utenti possano allo stesso modo cambiare l'avatar dell'utente che ha svolto la modifica nella loro scena.

3.3 Multiutente

Parte essenziale e centrale del lavoro svolto durante il tirocinio è stata proprio la possibilità di gestire più utenti all'interno di una vista in WebGL, con Funix per gestire questa importante feature si è scelto di adottare la tecnologia di WebRTC. WebRTC si tratta di una tecnologia OpenSource, ovvero un sistema di sviluppo software decentralizzato basato sulla condivisione del file sorgente. WebRTC permette di ottenere la comunicazione in tempo reale per il web, è quindi possibile realizzare funzionalità di invio dati video, vocali e generici tra le parti. In questa parte in particolare ci concentreremo sull'implementazione delle diverse funzionalità di multiutente ottenute grazie al canale dati arbitrari, quindi informazioni non multimediali, tramite il quale i diversi partecipanti possono:

- Comprendere quando entrano nuovi utenti nella stanza e quando escono
- Condivisione di informazioni utili all'identificazione quali UserID e Username
- Coerenza spaziale degli utenti quindi dove si trovano nella stanza
- Cambio avatar e notifica agli altri user nella stanza

3.3.1 WebRTC Tecnologia e Funzionamento

Come detto [WebRTC](#) si tratta di una tecnologia open source che permette la comunicazione in tempo reale per il Web, tramite questa è possibile aggiungere funzionalità di invio dati video, vocali e generici tra colleghi, permettendo di realizzare potenti soluzioni di comunicazione vocale e video. È disponibile su tutti i browser moderni, la tecnologia alla base di WebRTC sono open source e disponibili come API Javascript.



Figura 3.17. WebRTC Logo

Lo standard WebRTC copre a livello generale due tecnologie:

- Dispositivi di acquisizione di contenuti multimediali
- Connettività peer to peer

Quindi è possibile acquisire contenuti multimediali che includono videocamere e microfoni, ma anche acquisizione dello schermo. Per i primi quindi videocamere e microfoni si utilizzerà `navigator.mediaDevices.getUserMedia()` mentre per lo schermo `navigator.mediaDevices.getDisplayMedia()`. La connettività peer to peer (P2P) è

gestita dall'interfaccia `RTCPeerConnection`, si tratta del punto centrale per stabilire e controllare la connessione tra peers in WebRTC. Ora ci concentreremo sulla connettività P2P in quanto utile alla discussione delle diverse caratteristiche multiutente. Le connessioni peer sono la parte delle specifiche WebRTC che si occupa del collegamento di due applicazioni su computer diversi per comunicare tramite protocollo P2P. Due peer possono connettersi tramite dati audio, video o dati arbitrari, ovvero di qualsiasi entità, per connettersi è necessario che entrambi i client forniscano una configurazione `server ICE`. Questa può essere di tipo STUN o TURN il cui ruolo è quello di fornire candidati ICE a ciascun client che viene poi trasferito al peer remoto, trasferimento che viene definito come segnale. La specifica WebRTC include le API per la comunicazione con un server ICE (Internet Connectivity Establishment), si tratta quindi del protocollo per generare candidati di traversa media, è quindi necessaria la segnalazione tra le parti. In genere, si risolve tramite una normale API web basata su HTTP, ovvero un servizio REST (trasmissione di dati su HTTP senza ulteriori livelli) o tramite un altro meccanismo RCP (chiamata di procedure remota), in cui le applicazioni inoltrano le informazioni necessarie prima che venga avviata la connessione peer.

Ogni connessione peer viene gestita da un oggetto `RTCPeerConnection`. Il costruttore di questa classe prende come parametro un oggetto `RTCConfiguration`, che definisce la configurazione della connessione peer e contiene informazioni sul server ICE che si utilizzerà. Dopo aver creato l'elemento `RTCPeerConnection` viene creata un'offerta o una risposta dell'SDP (Session Description Protocol) a seconda che si tratti dell'app peer o del ricevente. Quindi una volta generata la segnalazione questa dovrà essere inviata al peer remoto tramite un canale diverso. Dopo aver impostato le descrizioni sia della sessione locale che della sessione remota, i due peer conoscono le funzionalità del peer remoto. Ciò non significa che la connessione tra peer è pronta. Per far sì che questo funzioni, è necessario raccogliere i candidati ICE per ogni peer e trasferirli (tramite il canale di segnalazione) all'altro peer. Prima che due peer possano comunicare utilizzando WebRTC, devono scambiarsi informazioni sulla connettività, per fare questo solitamente un servizio esterno per individuare i possibili candidati per la connessione a un peer. Come detto questo servizio è chiamato ICE e utilizza un server STUN o TURN. STUN indica un server Session Traversal Utility per NAT utilizzato indirettamente nella maggior parte delle applicazioni WebRTC. Con server TURN (Traversal Using Relay Nat) si indica la soluzione più avanzata che incorpora i protocolli STUN e la maggior parte dei servizi commerciali basati su WebRTC. L'API supporta direttamente STUN e TURN e viene raccolta con il termine più completo di definizione della connettività internet.

Dopo aver stabilito la connessione tra le parti la condivisione di informazioni che sia di streaming audio e video o di dati arbitrari. Per i primi sarà necessario creare uno stream che contenga almeno una traccia multimediale aggiunta singolarmente all'elemento `RTCPeerConnection` quando si vuole trasmettere l'elemento al peer remoto. Per inviare dati arbitrari invece si richiede la creazione di un canale dati che sempre sull'oggetto `RTCPeerConnection` restituisce un oggetto `RTCDataChannel`. Quindi il peer remoto ascolterà il canale tramite l'evento `datachannel` con la proprietà `channel` che rappresenta l'elemento connesso.

3.3.2 SkyWay Implementazione WebRTC nel progetto

Per implementare la tecnologia WebRTC nel progetto si è deciso di scegliere un servizio ovvero 'Enterprise Cloud WebRTC Platform' questo perché nell'ottica di future implementazione soprattutto della feature di comunicazione video sarà necessario, come visto in precedenza, appoggiarsi a server per la comunicazione.



Figura 3.18. SkyWay Logo

ECLWebRTC [Skyway](#) si tratta infatti di una SDK API che permette di aggiungere WebRTC nelle applicazioni, il vantaggio di questo servizio è che permette di utilizzare tramite API tutti i 4 server necessari per stabilire la comunicazione:

- Signaling Server
- STUN Server
- TURN Server
- SFU Server

Il Signaling Server permette lo scambio di informazioni necessarie per la comunicazione, come indirizzi IP e codecs, con una comunicazione tra i partner prima che la comunicazione venga stabilita.

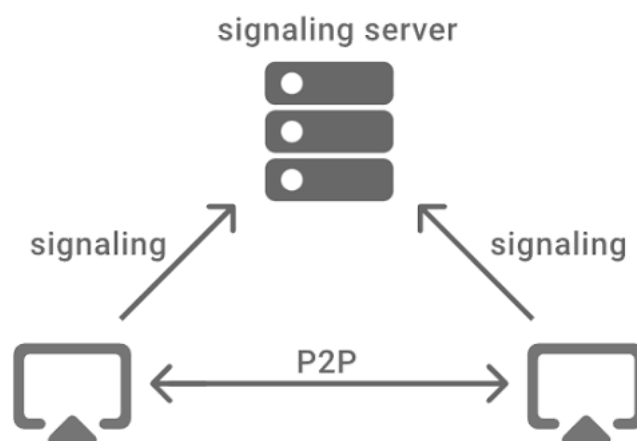


Figura 3.19. Comunicazione con il Signaling Server

Il server STUN (Session Traversal Utilities for NAT) permette di ottenere l'indirizzo IP globale ed il numero di porta dei terminali in ambienti di rete specifici in cui la comunicazione peer to peer non è possibile, come nelle reti aziendali.

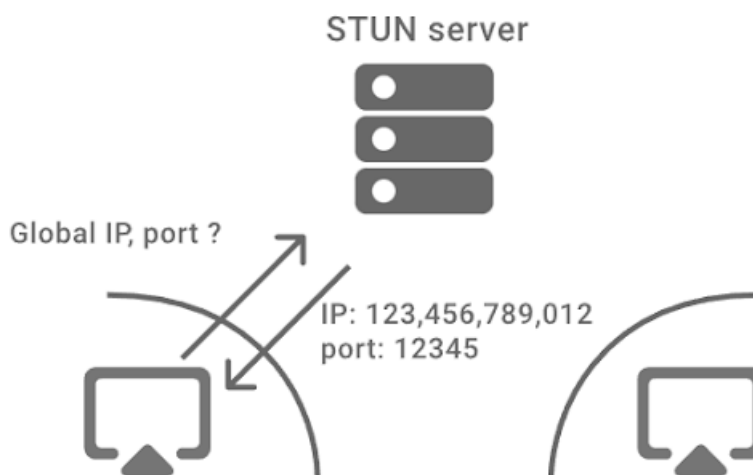


Figura 3.20. Comunicazione con il STUN Server

Il server TURN (Traversal Using Relays around NAT) permette di utilizzare WebRTC in ambienti di rete specifici dove la comunicazione peer to peer non è possibile, tramite lettura dei dati.

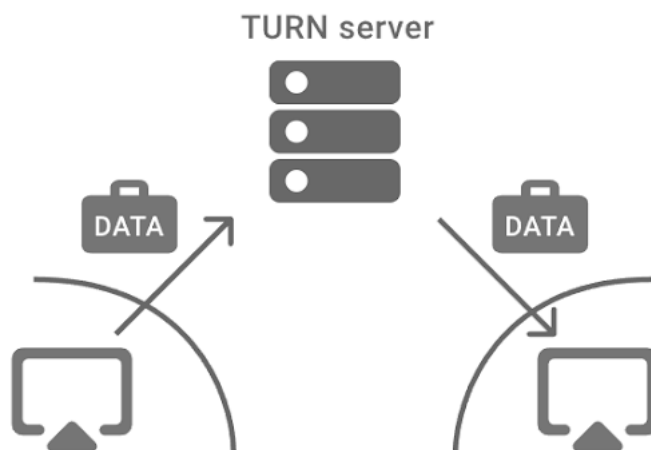


Figura 3.21. Comunicazione con il TURN Server

Il server SFU (Selective Forwarding Unit) realizza la comunicazione tra più persone e la possibilità di mostrare il video riducendo il carico di codifica, la larghezza di banda di uplink e il volume di comunicazione del terminale quando tre o più persone comunicano. Permettendo la comunicazione con più persone rispetto al metodo Mesh, ovvero un metodo che utilizza più comunicazioni P2P contemporaneamente decisamente poco efficiente.

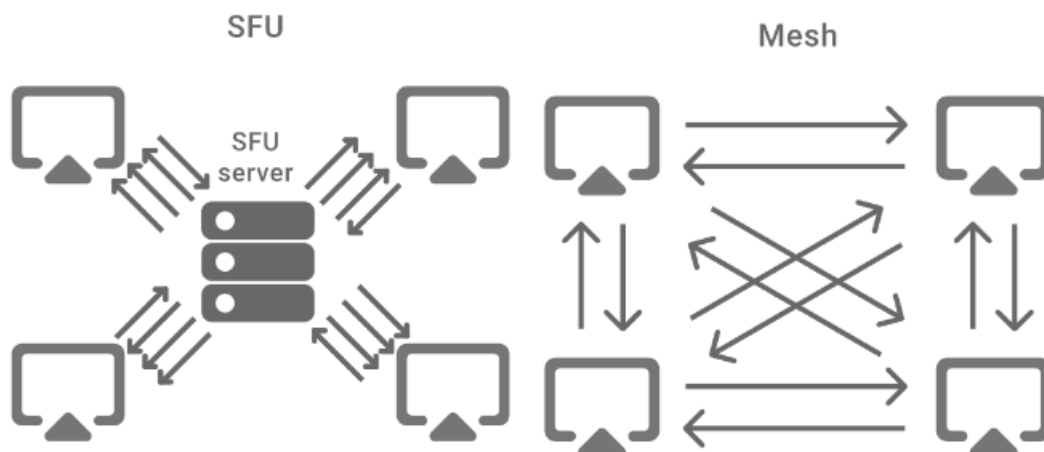


Figura 3.22. Cofigurazione SFU e Mesh dei peer

Nell'SDK fornita ci sono due metodi per la comunicazione il primo che permette una comunicazione 1:1 ed un'altra opzione definita come Room Model che permette a multipli peer di comunicare tra di loro come fossero in una conferenza video, questo è il modello implementato nella versione sviluppata. I peer entrano in una stanza specificando il nome della stessa, e se si riceve audio o video dagli altri partecipanti allora viene eseguito l'evento di stream e la comunicazione viene stabilita.

Implementare la SDK di Skyway per creare applicazioni web è tutto sommato semplice, una volta scaricata la SDK e quindi importati gli script necessari sarà necessario registrarsi al sito di Skyway per poter ottenere la API key. Si creerà quindi una nuova applicazione, dove dovrà essere aggiunto un dominio valido, e quindi si potrà creare l'applicazione ottenendo la API key, che ci servirà quando si crea il Peer. Per testare in locale tutte le funzionalità di WebRTC è necessario un web server, nel nostro caso è stato usato EasyPhp. Una volta eseguiti questi passaggi si può creare l'applicazione partendo dall'HTML e dallo script principale dentro il quale verranno impostati **peer**, **stanza** e **stream** dei vari partecipanti.

Passiamo quindi a descrivere il funzionamento del Multiutente all'interno del progetto. Ogni utente quando approda alla pagina della stanza potrà navigare liberamente esplorando in solitaria, nel frattempo il codice in Javascript si è già attivato per impostare tutto ciò che è necessario alla connessione, come lo stream delle tracce ed il peer, nel momento in cui l'utente decide di partecipare alla sessione gli basta cliccare sul tasto di Join ed inserire il proprio username, quindi sarà necessario verificare che il peer sia collegato al **signaling server**. Una volta verificata la connessione il peer entrerà nella stanza verranno eseguite tutte le funzioni di setup dell'utente:

- `UserConnectWhenLoad()`
- `SetBadge(...)`
- `SetUpUser(...)`

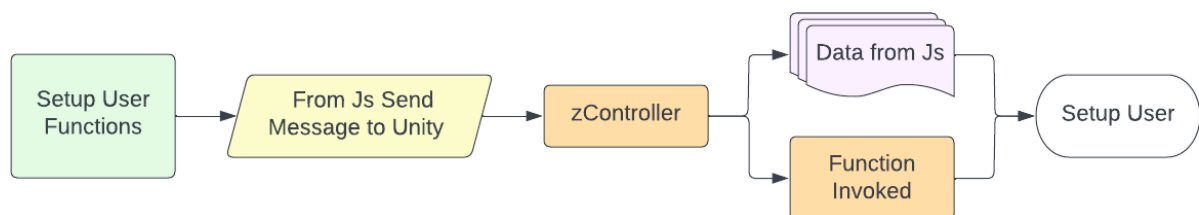


Figura 3.23. Diagramma delle funzioni di Setup

Quindi una volta connesso alla stanza verrà eseguita la funzione di connessione (`UserConnectWhenLoad`) la quale richiede a Unity tramite messaggio di istanziare l'utente local ed il suo modello con `Id` uguale al `peerId`. Le successive funzioni servono una per impostare il `Badge` che identifica ogni utente con l'username inserito, mentre l'altra più importante serve a segnalare a tutti gli altri utenti presenti nella stanza tutte le informazioni necessarie affinché l'entrata sia coerente con:

- Id dell'utente
- Username
- Posizione nello spazio
- Rotazione
- Cambio dell'avatar

Viene infatti creato un `Json`, ovvero un oggetto per trasportare e conservare informazioni. Dentro al quale verranno contenute tutte le informazioni. Una volta creato il `Json` questo viene mandato a tutte le viste `WebGL` tramite un evento che permette a Unity di comunicare con il lato `Javascript`, il quale recepisce il messaggio ed esegue un 'comando'.

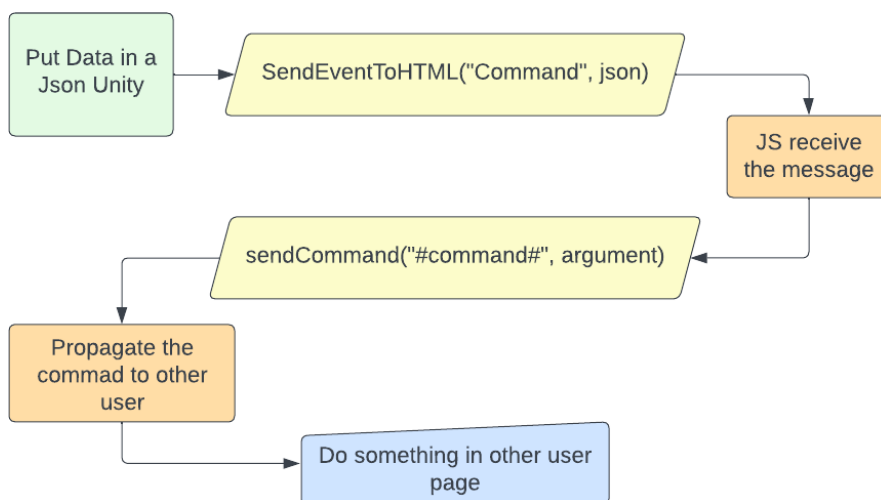


Figura 3.24. Diagramma di un HTML event che manda un comando ad altri utenti

Un comando non si tratta di altro se non di una funzione definita nello script principale che ci aiuta a comunicare tra le diverse istanze della stanza infatti ogni volta che viene invocato emette un messaggio nella stanza così che tutti coloro che sono all'interno possano sentire le informazioni che si stanno scambiando tutti gli utenti.

Quindi una volta entrato nella stanza verrà eseguito il comando `userSetup` che si bloccherà nel caso in cui la stanza dovesse essere vuota mentre eseguirà una nuova funzione nel momento in cui dovesse essere recepito da altri utenti: Quindi come visto la funzione `OtherUserConnectWhenLoad` eseguirà una funzione lato Unity così da rendere coerenti le diverse istanze viene infatti invocata la funzione `InstantiateNewUserJS` la quale, lato Unity, imposta gli utenti in base alle informazioni che lo stesso manda.

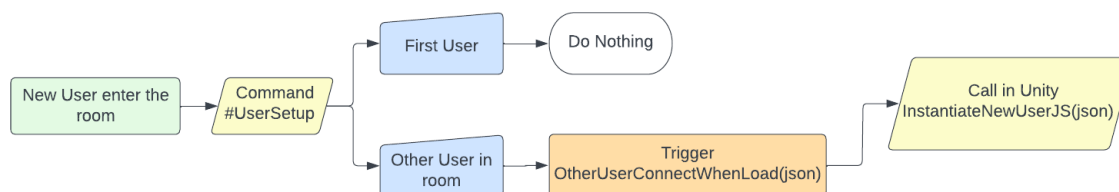


Figura 3.25. Diagramma di ciò che accade non appena un nuovo user entra in una stanza

Si verifica quindi che l'utente non sia già stato istanziato e nel caso in cui non lo fosse aggiunge un nuovo modello e gli imposta tutte le informazioni passate quali username, posizione/rotazione, e avatar personalizzato nel caso fosse stato impostato dall'utente.

Ogni altro utente lato unity verrà rappresentato da un `GameObject` definito come `'PlayerOther'` il quale è in tutto e per tutto simile a quello dell'utente principale a differenza della presenza della camera e dello script principale di movimento. Gli avatar degli altri utenti si muoveranno in scena ogni qual volta il proprio utente si muove nella scena triggerando l'evento `'UserPositionRotation'` questo dal Javascript esegue il comando che esegue lato Unity la funzione di `SetOtherUserJSON` la quale riceve come parametri la posizione raggiunta nello spazio e l'id dell'utente specifico, queste informazioni di posizione e rotazione vengono quindi passate al `GameObject PlayerOther` specifico dell'utente che grazie ad un `NavMeshAgent` permetteranno il movimento del modello nello spazio in maniera coerente in base alla posizione che deciderà l'utente di riferimento. In ultimo osserveremo il cambio dell'avatar, una volta cliccato il pulsante a schermo per l'attivazione di `ReadyPlayerMe` e una volta scelto l'avatar preferito o creato uno nuovo, verrà scaricato il modello questa operazione attiverà la funzione `'SwapCharacter'` la quale svolgerà tutte le operazioni di sostituzione e corretto posizionamento del modello precedente e cambio dell'animatore. Una volta svolte tutte queste operazioni verrà inviato un evento alla vista WebGL per informare tutti gli altri utenti del cambio tramite il messaggio `'UserChangedAvatar'` con allegate tutte le informazioni di URL del modello e ID specifico. Tramite il messaggio verrà eseguito il comando che attiva la funzione `'ChangedAvatar'` la quale eseguirà la funzione lato Unity `'SwapOtherCharacter'` questa attiverà l'avatarloader che modificherà similmente a quanto fa per l'avatar locale l'avatar dell'user specificato.

3.4 Comunicazione

In questo capitolo verrà discusso tutto ciò che riguarda la comunicazione tra utenti dalla chat di testo, alla trasmissione dei canali audio e video tramite webcam e la condivisione del proprio video all'interno di una scena in Unity. Verranno osservate quindi le tecnologie coinvolte ed il loro funzionamento alla base e quindi le integrazioni eseguite per permettere un corretto funzionamento delle parti per un'esperienza fluida e consistente. Verranno quindi discussi:

- Condivisione delle risorse multimediali grazie a WebRTC
- Integrazione del servizio Skyway per i media
- Condivisione Video e Audio tra utenti
- Condivisione schermo in scena

3.4.1 Media in WebRTC

Lo standard WebRTC fornisce API per l'accesso a videocamere e microfoni collegati al computer. Questi dispositivi sono comunemente chiamati dispositivi multimediali e sono accessibili con Javascript attraverso l'oggetto `navigator.mediaDevices`, che implementa l'interfaccia di `MediaDevices`. Da questo oggetto è possibile enumerare tutti i dispositivi connessi, rilevare le modifiche (connessi o disconnessi) e aprire un dispositivo per recuperare uno stream multimediale. Tramite la funzione `getUserMedia()`, si restituisce una promessa che si risolve in una `MediaStream` per i dispositivi multimediali corrispondenti, questa funzione utilizza un oggetto definito come `MediaStreamConstraints` che specifica i requisiti di cui disponiamo, per esempio aprire solo video e non microfono o aprire entrambi. La chiamata alla funzione attiverà una richiesta di autorizzazioni, se l'utente accetta allora la promessa viene risolta con una `MediaStream` contenente un video e una traccia audio. Nel caso l'autorizzazione dovesse essere negata si genera una promessa specifica così come nel caso non ci dovessero essere i dispositivi corrispondenti connessi. I vincoli che vengono definiti come parametro per la funzione `getUserMedia()`, permettono di aprire un dispositivo multimediale che soddisfa un determinato requisito, che può essere definito generalmente per audio e video o in maniera più precisa come risoluzione della fotocamera. I dispositivi saranno quindi configurati in base ai vincoli che possono essere appunto larghezza o altezze specifiche del video o la cancellazione dell'eco sui microfoni.

Nel momento in cui un dispositivo viene aperto si rende disponibile una `MediaStream`, questa verrà assegnata ad un elemento video o audio per riprodurre lo stream localmente ovvero ad elementi HTML definiti come video tag capaci di riprodurre quindi il video, i quali avranno parametri come `autoplay` o `playsinline` quest'ultimo permette allo stream video di essere riprodotto esattamente come definito piuttosto che a schermo intero. Le `MediaStream` come visto sono stream di contenuti multimediali che comprendono tracce diverse (audio e video), quando sono richiamate restituiscono un array di oggetti `MediaStreamTrack` che indica il tipo di media che rappresenta e che quindi ci permette di attivare o disattivare una traccia a piacere in base alle esigenze tramite le proprietà di `Enabled`.

3.4.2 Skyway e Condivisione video e audio tra utenti

Come visto precedentemente si è scelto per la creazione e la gestione di diverse feature come la comunicazione e la trasmissione di informazioni multimediali l'integrazione di una SDK e API fornita da ECLWebRTC, questo perché in grado di fornire server utili alla trasmissione di informazioni e alla gestione di più canali di comunicazione contemporaneamente mantenendo una consistenza e solidità.

L'implementazione di SkyWay WebRTC ci permette di avere delle stanze definite da un loro dominio e dati specifici come la API key per indirizzare la connessione dei diversi peer alla stessa. Quindi non appena un utente entra nell'applicazione tramite URL localmente verrà richiesta l'identificazione dei dispositivi multimediali tramite la definizione del `localStream`, verranno quindi definite le tracce ed i loro riferimenti nel codice HTML della vista WebGL, una volta identificati ed associate tutte le risorse verranno disabilitate così da permettere ad ogni utente di attivarle a piacere nel momento in cui si collega alla stanza.

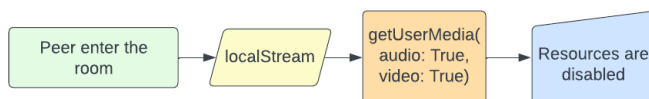


Figura 3.26. Diagramma di ciò che accade quando un Peer si collega

Successivamente verrà definito il Peer ovvero la classe che si comporta da entry point per la connessione alla stanza, infatti tra i parametri che definiscono il peer avremo la API key specifica dell'applicazione così che ogni peer possa collegarsi al peer remoto. Fintanto che il peer non verrà istanziato e verificato, l'utente non potrà connettersi alla stanza. Infatti l'utente potrà decidere quando entrare nella stanza condivisa e quando uscire. Quindi quando un utente decide di entrare nella stanza cliccando il pulsante 'Join', viene innescato l'evento `.joinRoom` che permette al peer di entrare nella stanza specificata dal nome, vengono forniti anche altri parametri come `roomMode` che identifica il tipo di configurazione scelta tra SFU o Mesh, e lo stream delle diverse tracce, se dovesse mancare questo valore si definisce il peer come solo capace di ricevere.

```

room = peer.joinRoom(roomName, {
  mode: roomMode,
  stream: localStream,
});
  
```

Figura 3.27. Valori funzionamento della stanza

Una volta accettato il peer e aperta la stanza, ovvero la stanza è pronta ed il Join è avvenuto con successo, verranno definite tutte le informazioni dell'user e degli altri come visto precedentemente, inoltre verrà aperta la possibilità all'utente di lasciare la stanza, di mostrare o meno il proprio video e di attivare o meno il microfono tramite pulsanti adibiti. Ogni volta che un nuovo peer entra nella stanza, quindi fornisce lo stream dei dati,

verranno creati per ogni scena i suoi tag HTML per contenere video e pulsanti specifici uno per ogni peerID. Una volta entrato l'utente potrà condividere il proprio video con gli altri oppure stopparlo se lo desidera.

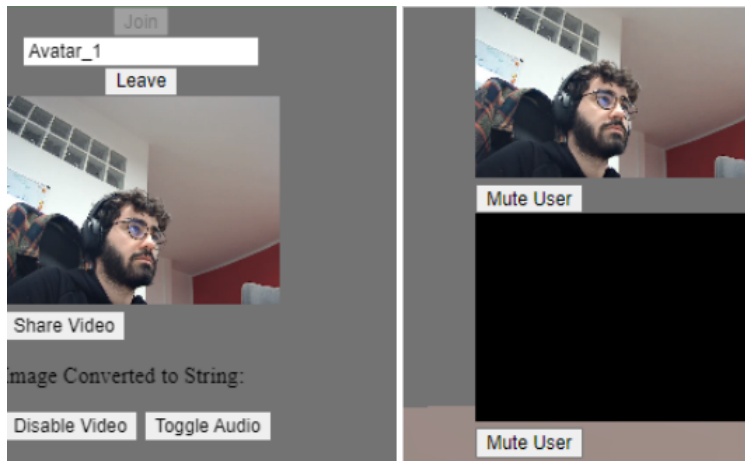


Figura 3.28. Tag video personale e degli altri utenti connessi alla stanza

Quando un utente decide di lasciare la stanza bisognerà gestire le sue risorse multimediali ed i tag HTML a lui connessi. Quindi nel momento in cui un altro utente abbandona la room bisognerà identificare il tag video legato al suo peerID, questo è possibile grazie ad un evento della libreria di Skyway, quindi se l'utente che abbandona la stanza stava condividendo lo schermo in scena gli viene interrotto lo stream, e successivamente vengono interrotte tutte le tracce per poi rimuovere il suo tag HTML, viene infine lanciato un messaggio a Unity che si occuperà di rimuovere l'avatar dell'utente ed i suoi riferimenti.

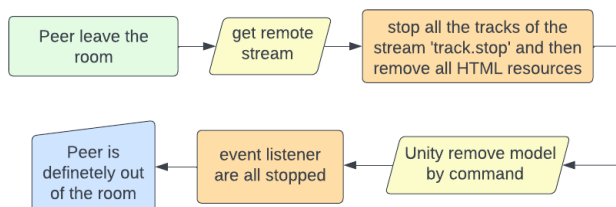


Figura 3.29. Diagramma di ciò che accade quando un Peer si scollega

Similmente quando l'utente principale decide di uscire dalla stanza verranno rimosse le risorse HTML per svolgere tutte le operazioni alla stanza, viene invocata la funzione per rimuovere gli altri utenti dalla scena Unity e se presente lo streaming video in scena verrà rimosso anche questo. Vengono infine rimossi tutti gli eventi associati ai pulsanti per evitare callback indesiderate ed in ultimo rimosse tutte le tracce video dagli elementi HTML altrui.

3.4.3 Condivisione Video

Ultima feature da analizzare è quella legata alla possibilità di condividere il proprio video in scena. Ogni utente a quando entra nella stanza potrà condividere il proprio video in scena per condividerlo internamente nello spazio 3D, se viene fatto da qualcun altro non sarà possibile condividere lo schermo. Quando l'utente vorrà condividere il video verrà quindi verificato lo stato della condivisione, se attiva o meno, e l'id di chi condivide lo schermo per quindi ottenere le informazioni dal canvas contenente il video quali altezza e larghezza.

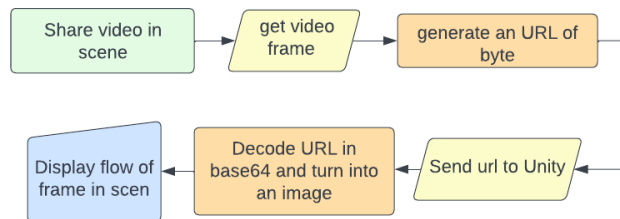


Figura 3.30. Diagramma degli eventi di condivisione schermo

Da queste informazioni si ricaverà un URL contenente i byte di informazioni delle immagini del video che verranno mandati alla scena Unity. Una volta arrivate nella scena Unity verrà decodificato l'URL da base64 ad un'immagine che verrà proiettata in scena, questa operazione verrà ripetuta continuamente fintanto che l'utente decide di condividere il video in scena creando quindi un flusso di immagini che genera il video. Se nessuno condivide lo schermo sarà presente un'immagine di default.



Figura 3.31. Condivisione del video in scena

Capitolo 4

User Test

4.1 Raccolta

L'implementazione del multiutente per l'applicazione di Virtual Expo è stato sottoposto a test da parte dell'azienda e da alcuni utenti compatibili con i target di riferimento, quindi utenti con competenze e conoscenze minime di come ci si muove su una pagina web e di come si utilizzano alcuni strumenti tecnologici quali webcam e microfoni.

I tester sono stati posti di fronte ad un utilizzo delle feature in piena libertà come fossero approdati sul portale di Virtual Expo con tutte le feature sviluppate, gli è stato quindi fornito il link all'applicazione in WebGL dando libertà di movimento nella stanza e quindi libertà di decidere se partecipare o meno alla sessione condivisa in piena libertà. Si sono quindi osservate le reazioni, ascoltati i commenti e analizzato il flusso di interazione degli utenti necessario per comprendere se si sono pensate o meno correttamente le diverse parti e nel caso di correggerle negli sviluppi futuri del progetto. Dall'analisi emergono quindi spunti di vario tipo, conferme su incertezze di sviluppo e di implementazione e quindi idee per nuove feature.

Durante i test i gruppi di utenti hanno massima libertà e non sono guidati, la mia figura di super user era presente solamente per poter osservare i comportamenti all'interno della scena ma distaccata proprio per avere massima trasparenza nei commenti raccolti.

4.2 Test Aziendali Interni

I test di maggior interesse per il lavoro e svolti frequentemente durante il lavoro di tirocinio riguardano quelli interni svolti con i diversi colleghi in ufficio, quindi con persone ad alta competenza informatica e con capacità di comprendere meglio le criticità di alcune scelte rispetto ad altre. Con i componenti dell'ufficio quindi si sono svolti diversi test principalmente due sono da denotare il primo a metà lavoro circa ed un secondo test svolto in fase avanzata del progetto.

Nel primo test è importante sottolineare che la build in WebGL del progetto era ancora implementata con una soluzione custom di WebRTC, era quindi una versione nella quale tutte le funzioni di comunicazione e di integrazione del protocollo erano state create da

zero. Questo da subito ha portato ad alcune criticità infatti non appena i colleghi hanno avuto piena libertà di muoversi e testare le diverse feature sono emerse alcune importanti problematiche:

- Mancanza di feedback visivi per le operazioni svolte
- Interfaccia povera e poco esplicativa
- Problemi di latenza se stressata la stanza con l'entrata di più utenti
- La gestione di più utenti e quindi l'appesantimento dei canali di comunicazione impedisce una corretta comunicazione e il presentarsi di alcuni bug rende poco stabile l'applicativo
- Gestione complicata delle risorse multimediali

Nonostante le problematiche sono emersi anche pregi come:

- Rapidità del collegamento alla stanza
- Semplicità di navigazione all'interno della stanza e precisione dei movimenti dei diversi utenti
- Ottima la feature di personalizzazione e l'integrazione di Ready Player Me

Dopo questi test preliminari ma importanti si è deciso di cambiare approccio alla gestione di più utenti all'interno di una stanza e studiando meglio il problema un approccio totalmente custom si è rivelato difficile e si è preferito passare all'implementazione di una SDK per diverse ragioni quali:

- Comunicazione Full Mesh, poco ottimale siccome ogni peer è appesantito dal dover stabilire una connessione con ogni membro
- Gestione delle risorse multimediali difficile
- Canali di comunicazione sia dei dati che degli stream rallentata
- Gestione di un Signaling Server Complessa
- Necessità nel futuro di dover aggiornare tutta la struttura del codice, con un SDK risulta tutto molto più semplice

Prese in considerazione queste criticità è stata implementata la SDK SkyWay di 'Enterprise Cloud webRTC' come visto nel capitolo dedicato, questa decisione ha permesso di risolvere molti dei problemi evidenziati di sopra e di ridurre altre problematiche.

Quindi è stato svolto un secondo test interno al gruppo di lavoro in una fase in cui la build era decisamente più solida grazie all'implementazione del SDK, inoltre in questa fase di test sono state implementate altre feature quali la possibilità di entrare ed uscire dalla stanza senza dover chiudere la pagina e la possibilità di condividere lo schermo all'interno della scena. Durante questo test i colleghi hanno da subito notato diverse migliorie quali:

- Feedback visivi per alcune azioni migliorati
- Un'interfaccia più esplicativa rispetto a prima ma ancora da migliorare essendo un template
- Capacità di reggere maggiormente gli stress con più utenti, fino a 20 contemporaneamente con condivisione delle risorse multimediali grazie alla nuova configurazione della comunicazione
- Correzione di bug relativi, grazie alla solidità delle diverse funzioni offerte dal SDK
- Miglioramento della gestione delle risorse multimediali, molto più semplice intuitiva e personalizzata

D'altra parte sono comunque emerse alcune criticità durante i test dovute principalmente ad una scarsa cura dei contenuti visivi avendo prediletto le funzionalità tecniche e la consistenza della comunicazione tra le parti. Quindi in questa seconda fase di test sono state criticate:

- Interfaccia poco curata
- Alcuni scatti non impediti dovuti al download dei modelli di Ready Player Me
- Abbassamento degli fps non deleterio, quando si condivide lo schermo in scena

4.3 Test Gruppo di Studenti

In questo altro test svolto è stato chiesto ad un gruppo eterogeneo di 5 studenti universitari di testare l'applicativo e le funzionalità tramite una chiamata di discord per avere supporto durante le operazioni. Una volta riuniti ho condiviso loro il link della build WebGL e li ho lasciati liberi di esplorare l'ambiente virtuale mutando i partecipanti alla stanza discord. Mentre gli utenti esploravano la stanza io stesso ho partecipato alla loro esperienza come utente seguendo a mia volta il link. Il test quindi voleva principalmente comprendere la facilità d'uso, la stabilità del sistema di comunicazione e l'interesse verso questo tipo di tecnologia ed esperienze.

Quindi non appena entrati gli studenti hanno potuto esplorare in autonomia la stanza, infatti fintanto che un utente non esegue il Join alla stanza sarà libero di esplorare la stanza in solitaria. Dopo qualche minuto gli utenti hanno iniziato ad entrare nella stanza condivisa e ad attivare le diverse risorse multimediali iniziando a stabilire una comunicazione e un'esplorazione completa. Una volta terminata l'esperienza ho richiesto dei feedback partendo dalla facilità d'uso qui riassunti alcune osservazioni comuni o di rilevanza:

- L'esplorazione all'interno della stanza virtuale è intuitiva e libera
- Il doppio sistema di movimento confonde inizialmente ma permette di muoversi come più si preferisce

- L'interfaccia per quanto efficace risulta ancora acerba
- Entrare nella room condivisa per quanto facile risulti manca di feedback
- Cambiare avatar è intuitivo, veloce ed il tool implementato ha un buon livello di personalizzazione
- La chat testuale fornisce alcuni feedback importanti che altrimenti non vengono segnalati
- Quando un nuovo utente entra nella stanza non viene segnalato in alcun modo se non nella chat
- Comunicare tramite i canali audio e video è semplice e funzionale tuttavia manca l'audio spazializzato quindi gli utenti si confondono soprattutto per la posizione nello spazio
- Condividere il proprio video in scena è una feature utile e interessante tuttavia servirebbe gestire meglio la possibilità di richiedere questo permesso o di concederlo, potendolo fare un solo user per volta.

Quindi dopo aver raccolto i primi feedback sull'esperienza ho chiesto pareri sulla parte più corposa e di mio interesse nello sviluppo con WebRTC ovvero la stabilità del sistema e la consistenza della comunicazione tra le parti.

- Il sistema appare solido, non sono apparsi glitch grafici o bug distruttivi
- La consistenza dei diversi utenti è ben gestita, gli avatar si muovono e condividono tra le parti le esatte coordinate spaziali e le rotazioni
- La comunicazione video e audio è ben gestita, l'audio risulta pulito e comprensibile così anche i video dei diversi partecipanti
- Chi condivide lo schermo in scena lamenta alcuni cali di frame che comunque non risultano compromettenti per l'esperienza
- Il cambio degli avatar risulta efficace qualche leggero scatto nel momento in cui l'avatar viene istanziato tuttavia anche questo non compromette l'esperienza

Ultima domanda posta ai tester riguardo all'esperienza riguarda a future implementazioni di questa tecnologia e ad esperienze simili se ne hanno avute in precedenza, quindi se hanno un interesse verso il concetto di metaverso. Ecco alcune delle risposte raccolte più importanti:

- L'esperienza è interessante nell'ottica di piccole applicazioni o eventi siccome certamente rende più coinvolgente e personale l'esperienza
- La tecnologia non è certamente innovativa tuttavia in un contesto di ambienti virtuali si rivaluta e le applicazioni che la integrano diventano esperienze più immersive

- L'idea sviluppata nell'ottica di far conoscere le aziende in un modo differente è interessante, siccome accorcia la distanza tra azienda e utente finale
- L'applicazione provata è semplice e acerba, però se migliorata e implementata per le giuste aziende potrebbe essere un buon modo di sponsorizzare le stesse
- In precedenza ho provato applicazioni simili a questa con scopi diversi come gallerie virtuali, l'idea che si possano provare applicazioni simili per realtà aziendali sicuramente è un'ottima pubblicità se ben implementata
- Nonostante non abbia mai provato applicazioni simili tuttavia il concetto di metaverso mi affascina anche se ritengo che la tecnologia attualmente è acerba
- Il metaverso e le sue realizzazioni attuali sono acerbe e poco pratiche, spesso fini a loro stesse. Inoltre trovo che l'espressione del concetto tramite il mondo delle criptovalute e degli NFT per come è ora sviluppato sia decisamente corrotto solamente dall'idea del guadagno.

Capitolo 5

Conclusione e Sviluppi futuri

L'obiettivo del lavoro durante il tirocinio e quindi per la tesi è stato quello di sviluppare delle feature di multiutente per il progetto di Virtual Expo dell'azienda Funix. Il lavoro svolto ha permesso di sviluppare un prototipo di applicazione, con le feature richieste di avatar modificabili, multiutente e comunicazione multimediale. Grazie alla tecnologia di WebRTC è stato possibile sviluppare le principali feature che vogliono simulare il concetto di Metaverso molto allettante e richiesto dalle aziende. Tuttavia il mercato che ruota attorno al metaverso è ancora acerbo, spesso le piccole aziende ci si tuffano per tendenza e faticano loro stesse a comprendere i rischi e le difficoltà tecniche. Funix per questo ha deciso di provare ad avvicinare le piccole realtà al metaverso, vista la crescente richiesta tramite il progetto di VirtualExpo e le feature sviluppate di multiutente.

Il progetto di VirtualExpo vive indipendentemente dalle feature sviluppate le quali si sono funzionanti ma ancora non implementate dall'azienda. Nel futuro sarà quindi necessario un lavoro più preciso su diversi aspetti. Primo fra tutti sarà necessario implementare le feature sviluppate così da rendere possibile l'interazione tra utenti all'interno dell'applicazione. Sarà inoltre necessario migliorare l'interfaccia utente visti i commenti degli user test, sono necessari pochi accorgimenti mancanti durante la fase di sviluppo in favore delle funzionalità principali di multiutente. L'integrazione con SkyWay andrà inoltre implementata per poter avere la massima capacità da parte dei loro server, siccome attualmente le funzionalità sono implementate con le risorse da sviluppatore. Sarà inoltre necessario ottenere la partnership con il tool di Ready Player Me così da poter sfruttare le loro funzionalità in regola e non in modalità sviluppatore. Le funzionalità attuali permettono agli utenti di esplorare una sola stanza ma nel futuro si potrebbe decidere di implementare dei portali su più luoghi così da creare un metaverso collegato. Con le modifiche sopra citate si potrebbe ottenere una versione di Virtual Expo molto più improntata alla condivisione con la possibilità di collegare più utenti in ambienti inerenti a piccole realtà, così da avere dei luoghi dentro i quali le aziende potrebbero mostrare ambienti virtuali capaci di rappresentare la loro realtà secondo le loro esigenze, ma potrebbero anche interagire con la loro utenza accorciando le distanze rendendo la comunicazione innovativa e coinvolgente.

Bibliografia

Agarawala Anand and Lee Jinha. Spatial url: '<https://www.spatial.io>'. 2022.

Sébastien Borget. The sandbox (sand): Tokenized assets for gaming ecosystems. 2021.

Keshab Nath. Evolution of the internet from web 1.0 to metaverse: The good, the bad and the ugly. 2022.

RPMdoc. Ready player me documentation : <https://docs.readyplayer.me/ready-player-me/>.

Skyway. Skyway documentation : <https://webrtc.ecl.ntt.com/en/documents/javascript-sdk.html>.

Neal Stephenson. *Snow Crash*. 1992.

Hollensen Svend, Kotler Philip, and Opresnik Marc Oliver. Metaverse – the new marketing universe. *Journal of Business Strategy*, 2022.

Paul Tassi. Fortnite's epic games makes a metaverse investment to scale up even further. 2022.

Jung Timothy and tom Dieck M. Claudia. Augmented reality and virtual reality empowering human, place and business. 2018.

UnityCam. Cinemachine documentation : <https://docs.unity3d.com/packages/com.unity.cinemachine/manual/>.

UnityNavmesh. Navmeshagent documentation : <https://docs.unity3d.com/manual/class-navmeshagent.html>.

WebRTC. Webrtc documentation : <https://webrtc.org/getting-started/overview>.

Hater Wes and Hempton Gordon. Spot url: '<https://www.spotvirtual.com/>'. 2022.