

POLITECNICO DI TORINO

Collegio di Ingegneria Informatica, del Cinema e Meccatronica

**Corso di Laurea Magistrale
in Ingegneria Informatica**

Tesi di Laurea Magistrale

Sviluppo di un'Applicazione di Realtà Virtuale Immersiva per l'esplorazione dell'Universo



Relatore

prof. Andrea Sanna

Candidata

Natalia Lanza

Marzo, 2020

ABSTRACT

Il progetto di tesi si colloca nell'ambito della realtà virtuale e si pone come obiettivo il miglioramento e l'estensione dello strumento Astra Data Navigator. Scopo del progetto precisamente era la realizzazione di un'applicazione che consentisse un'esplorazione realistica e precisa di cataloghi astronomici in un ambiente 3D virtuale che fosse navigabile e interattivo.

Il software è stato realizzato presso l'azienda ALTEC - Aerospace Logistics Technology Engineering Company - centro di eccellenza italiano per la fornitura di servizi di ingegneria e logistica a supporto delle operazioni e dell'utilizzo della Stazione Spaziale Internazionale e dello sviluppo e supporto di missioni di esplorazione planetaria.

Gli oggetti astronomici, contenuti in specifici file, vengono caricati dinamicamente durante l'esecuzione dell'applicazione. L'applicazione consente di visualizzare gli oggetti caricati con possibilità di ricavare nuove informazioni e conoscere meglio le loro caratteristiche. L'utente, attraverso questo software può infatti scegliere su quale oggetto soffermarsi.

Per realizzare il lavoro si è scelto di utilizzare la piattaforma di sviluppo 3D in tempo reale di Unity, tuttavia, nonostante questa sia uno strumento semplice e utilizzabile per qualsiasi tipo di applicazione, per simulare un ambiente in scala astronomica non si è dimostrata essere l'ideale a causa delle limitazioni dell'API e dei tipi di dati offerti.

Per superare i limiti presentati e rendere l'applicazione più precisa e realistica possibile sono stati quindi sviluppati appositi meccanismi di gestione. Sono stati analizzati e sono stati realizzati diversi approcci tradizionali, quali l'utilizzo di sistemi di coordinate locali multiple, il concetto di scale space e la tecnica del floating origin.

INDICE

INTRODUZIONE	1
Il ciclo di Hype	1
Il mercato della realtà virtuale	2
Il progetto di tesi	2
CAPITOLO 1	4
Stato dell'arte	4
1.1 La NASA	4
1.1.1 Chandra X-ray Center: la missione principale della NASA per l'astronomia a raggi X	5
1.2 Educazione astronomica e realtà virtuale	6
1.3 Applicazioni per esplorare l'universo	8
1.4 Altec: Astra Data Navigator	9
1.5 La nuova versione di Astra Data Navigator	9
CAPITOLO 2	11
La Realtà Virtuale	11
2.1 Cos'è la realtà virtuale	11
2.2 Tipi di Realtà Virtuale e concetti base	12
2.3 Stereoscopia	14
2.3.1 Sistemi multiplexed	15
2.4 Realtà Virtuale: passato e presente	17
2.5 Limiti della realtà virtuale	19
2.6 La realtà virtuale nel settore aerospaziale	19
2.6.1 La realtà virtuale nello studio astronomico	19
2.7 Realtà virtuale in Altec	20
CAPITOLO 3	22
Tecniche di realizzazione	22
3.1 Virgola Mobile	23
3.2 Floating Origin	24
3.3 Scale Space	25
3.3.1 Scale Space Object	26
3.3.2 Scale Space Camera	26
3.3.3 Scale Space Manager	26
3.4 Asset Space Graphics Toolkit	26
3.4.1 SgtFloatingPoint	27

3.4.2 SgtFloatingCamera.....	27
3.4.3 SgtFloatingOrigin.....	27
3.4.4 SgtFloatingScaler	27
Capitolo 4	28
Progettazione dell'applicazione	28
4.1 Catalogo Hipparcos	28
4.1.1.2 Stella 3D.....	29
4.1.1.3 Particelle.....	30
4.1.2 Descrizione della stella.....	31
4.1.3 Selezione Target (e spostamento)	32
4.2 Sistema Solare	33
4.2.1 Modelli	34
4.2.2 Il moto	34
4.2.3 Selezione Pianeta.....	35
4.2.4 Descrizione dell'oggetto astronomico.....	35
4.3 Navigazione.....	36
4.4 GUI.....	36
4.4.1 Elementi di interazione.....	37
4.5 Stereoscopia	37
Capitolo 5	39
Criticità e Test	39
5.1 Criticità.....	39
5.1.1 Caricamento e lettura del file del catalogo Hipparcos	39
5.1.2 Gestione del numero elevato di stelle	41
5.2 Test di valutazione soggettivi.....	42
CAPITOLO 6	46
Conclusioni e Sviluppi futuri	46
APPENDICE	49
A.1 FloatingScaleParticle: ResizeSystem	49
A.2 ParticleGenerator: Add.....	49
A.3 FloatingScaleParticle: ResizeParticle.....	50
A.4 StarInfoPanel	50
A.5 FloatingScalerPlanet: SetDistance	51
A.6 Questionario per la valutazione di Astra Data Navigator.....	51
Bibliografia.....	54
RINGRAZIAMENTI	55

Elenco delle figure

Figura 0.1 – VR Hype Cicle.....	1
Figura 0.2 - Settori in cui le tecnologie XR potranno essere di rilevante importanza.	2
Figura 1.1 - Creazione della prima stampa 3D in assoluto di un residuo di supernova basato sui dati osservati.....	6
Figura 1.2 - Esempio di Space Engine.	8
Figura 1.3 - Esempio di Celestia.	9
Figura 2.1 - Il senso di presenza aumenta all'aumentare del senso di immersione e interazione.	12
Figura 2.2 - A sinistra un esempio di realtà virtuale non immersiva, a destra un esempio di realtà virtuale immersiva.....	13
Figura 2.3 - Esempio di filtro polarizzatore.	16
Figura 2.4 - Esempio di anaglifi.....	17
Figura 2.5 - Esempio di gamma di colore.	17
Figura 2.6 - Esempio di dati stellari di Gaia in PointCloudsVR.	20
Figura 2.7 - VR Room di Altec.	21
Figura 2.8 - Configurazione mobile del sistema di supporto alla VR in Altec.	21
Figura 3.1 - Il frustum di visualizzazione è la regione dello spazio nel mondo modellato che può apparire sullo schermo. Gli oggetti più vicini alla telecamera rispetto al piano di clipping frontale o oltre il piano di clipping lontano non vengono disegnati.....	22
Figura 3.2 - Esempio rappresentazione di un numero secondo lo standard IEEE 754.	24
Figura 3.3 - Confronto tra il tradizionale movimento del punto di vista con il meccanismo del floating origin.	24
Figura 3.4 - Struttura del client in un sistema distribuito: mostra come il punto di vista centralizzato e i componenti di navigazione influenzano il WT che, a sua volta, trasforma gli oggetti. Una volta posizionati gli oggetti attivi, i delta di navigazione inversa inviati al framework degli oggetti attivi mantengono aggiornate le loro posizioni.	25
Figura 3.5 - Meccanismo scale space.....	25
Figura 4.1 - Diagramma di flusso della generazione della scena.....	29
Figura 4.2 - Pannello del Sistema Particellare in Unity.	30
Figura 4.3 - Esempio dei modelli adoperati per le stelle: sfera e particelle.	31
Figura 4.4 – Componenti da aggiungere al pannello e alla camera affinché il movimento di trascinamento funzioni correttamente.	32
Figura 4.5 - Immagine raffigurante la Terra con i suoi satelliti.	34
Figura 4.6 - Ricerca dell'oggetto astronomico tramite barra di ricerca.....	35
Figura 4.7 - Immagine della GUI di Astra Data Navigator.....	37
Figura 5.1 – Caricamento e lettura del file del catalogo Hipparcos caso 1.....	40

Figura 5.2 - Caricamento e lettura del file del catalogo Hipparcos caso 2.	41
Figura 5.3 – Valutazione soggettiva del carico di lavoro mentale.	43
Figura 5.4 – Grafico per la valutazione dell'intuitività.	43
Figura 5.5 – Grafico per la valutazione dell'usabilità.	44
Figura 5.6 – Grafico per la valutazione del realismo.	44
Figura 5.7 – Grafico per la valutazione del senso di nausea.	45
Figura 5.8 – Grafico dei suggerimenti futuri per in miglioramento di Astra Data Navigator.	45

Elenco delle tabelle

Tabella 1.1 - Alcuni sistemi VR desktop utilizzati in ambito educativo.....	7
Tabella 4.1 – Caratteristiche di una stella.	29
Tabella 5.1 - Intervallo di tempo medio ottenuto durante le simulazioni caso 1.	40
Tabella 5.2 - Caricamento e lettura del file del catalogo Hipparcos caso 2.	40

INTRODUZIONE

Un tempo la realtà virtuale era qualcosa che era possibile solo immaginare ma oggi è diventata una realtà vera e propria. Secondo i dati elaborati da Gartner [1] la realtà virtuale nel 2018 ha smesso di “avere potenzialità” e ha iniziato ad essere reale. Il 2019 rappresenta l’anno di svolta in cui la realtà virtuale diventa una tecnologia matura e inizia ad occupare la fase finale del ciclo di Hype (figura 0.1).

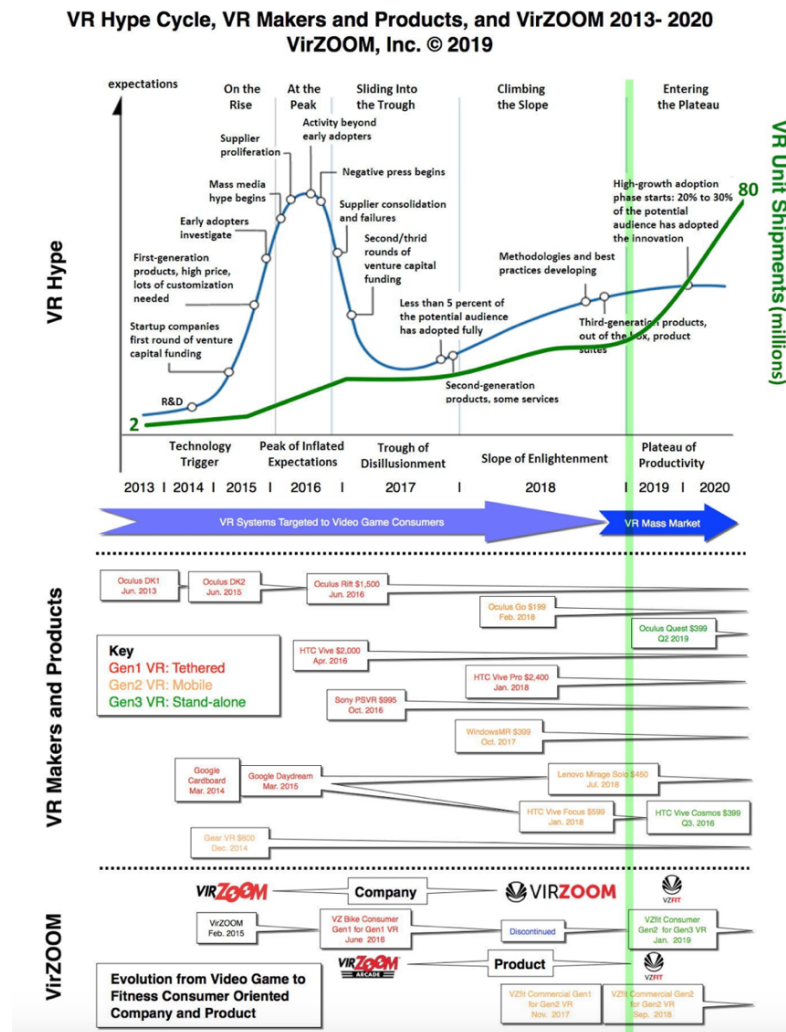


Figura 0.1 – VR Hype Cycle.

Il ciclo di Hype

Il ciclo di Hype è una rappresentazione grafica che serve ad illustrare il ciclo di vita di una tecnologia, aiuta a stimarne la crescita o la fine e si compone di 5 fasi:

- **Innesco della tecnologia:** è la fase iniziale in cui si cerca di analizzare l’interesse delle persone per il nuovo prodotto pur non essendo esso ancora disponibile sul mercato.
- **Il picco delle aspettative gonfiate:** la tecnologia inizia ad essere presente sul mercato e viene prestata molta attenzione alla pubblicità. Questa fase è seguita da un declino dovuto al fatto che non si raggiungono i risultati promessi, gli utenti rimangono delusi e il loro interesse svanisce.

- **Disillusione:** le aziende che affrontano i problemi e continuano ad investire lanciano sul mercato nuovi prodotti che hanno l'obiettivo di dimostrare le potenzialità di funzionamento della nuova tecnologia.
- **“Pendenza” di miglioramento:** le aziende definiscono i “prodotti basati sulla tecnologia futura” e li testano nell'ambiente reale. Innovazione e investimenti sono sempre più importanti.
- **Aumento della produttività:** i prodotti si diffondono e i produttori stabiliscono i nuovi standard di qualità.

Il mercato della realtà virtuale

Gartner, dopo aver elaborato i dati ottenuti durante la ricerca “The future of immersive experience”, ritiene che, entro il 2022, le tecnologie immersive saranno state sperimentate dal 70% delle aziende mondiali e il 25% le avrà adoperate a fini produttivi.

Diversi sono i settori in cui l'impiego della tecnologia VR¹ sta crescendo. Una notevole diffusione si registra nel settore logistico e delle spedizioni, nella diagnosi medica, e nell'educazione basata sullo sviluppo di ricerche in laboratorio e sul campo. Uno studio effettuato da Accenture vuole sottolineare come l'impiego di tecnologie XR² può potenzialmente aumentare la produttività (figura 0.2) [2].

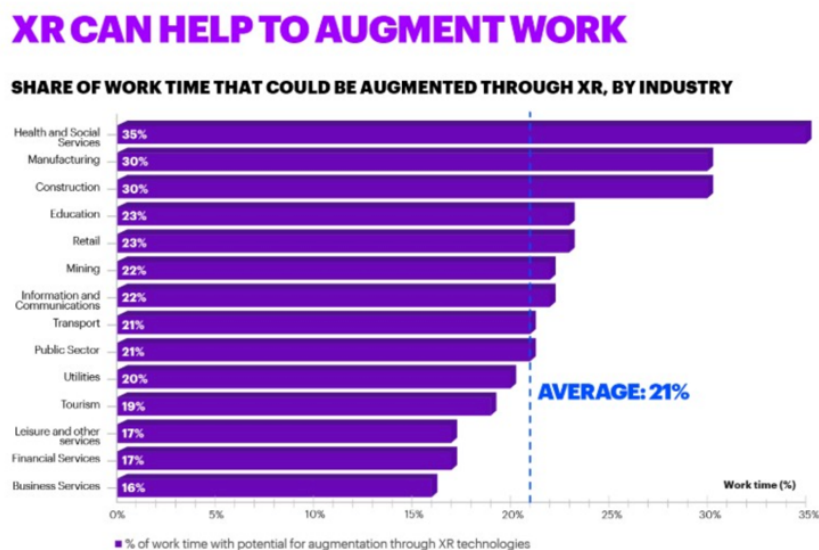


Figura 0.2 - Settori in cui le tecnologie XR potranno essere di rilevante importanza.

Il progetto di tesi

Il progetto di tesi rappresenta un esempio di applicazione di realtà virtuale potenzialmente sfruttabile nel campo dell'educazione e nel settore astronomico per la visualizzazione di dati scientifici. Questa applicazione vuole dare la possibilità di osservare i dati ottenuti dai satelliti roaming Hipparcos e la possibilità di esplorare il sistema solare per avere nuove informazioni

¹ VR: Realtà virtuale

² XR comprende realtà virtuale (VR), realtà aumentata (AR), tattili, ologrammi e una gamma in espansione di strumenti immersivi che utilizzano e migliorano i nostri sensi naturali.

e conoscere le caratteristiche dei corpi celesti, vuole dunque permettere l'esplorazione dell'universo osservabile.

La tesi è composta da sei capitoli, di seguito ne viene fornita una breve descrizione.

Il primo capitolo valuta l'impatto della realtà virtuale nel settore astronomico. Vengono presentati alcuni progetti in fase di sviluppo della NASA, viene analizzata la VR desktop come mezzo di aiuto per l'insegnamento dell'educazione astronomica e vengono descritte alcune applicazioni utilizzabili per la rappresentazione e l'esplorazione dell'universo. Infine viene fatto un breve confronto tra la vecchia versione dell'applicazione posseduta da Altec e la nuova versione ampliata e sviluppata con questo progetto di tesi.

Il secondo capitolo definisce il concetto di realtà virtuale. Vengono analizzate le caratteristiche chiave e le diverse tipologie di questa tecnologia, viene presentata la sua evoluzione e viene approfondito il suo uso nel settore aerospaziale con particolare attenzione rivolta allo studio astronomico.

Il terzo capitolo tratta le possibili tecniche di realizzazione applicabili per rimediare al problema della perdita di precisione che si verifica nel momento in cui si vuole rappresentare uno spazio molto grande come l'universo. Infatti il software Unity per memorizzare la posizione di un oggetto utilizza il formato a virgola mobile a precisione singola Vector3 (float) che però può portare ad una perdita di precisione quando l'oggetto si trova molto distante dall'origine. Sarebbe opportuno ricorrere all'utilizzo di un Vector3 in doppia precisione (double) ma questa opzione non è disponibile.

Il quarto capitolo illustra le fasi di progettazione e sviluppo dell'applicazione. Vengono analizzati nel dettaglio i vari componenti, vengono descritti gli elementi grafici e le funzionalità, e vengono definite le proprietà che occorre impostare affinché l'applicazione possa funzionare correttamente in modalità stereoscopica.

Il quinto capitolo presenta le problematiche e le criticità riscontrate durante lo sviluppo. Viene inoltre rappresentata un'analisi statistica di test soggettivi effettuati riguardanti l'uso, il realismo e l'intuitività del sistema, nonché il senso di nausea provocato dall'uso dell'applicazione in modalità stereoscopica.

Il sesto capitolo espone le conclusioni e fornisce dei suggerimenti per possibili sviluppi futuri dell'applicazione.

CAPITOLO 1

Stato dell'arte

La realtà virtuale permette di simulare la realtà, è descritta come un ambiente multimediale interattivo in cui l'utente diventa partecipe del mondo generato dal computer. Il partecipante naviga nel mondo realizzato e interagisce con gli oggetti presenti in esso in tempo reale grazie al computer che a seguito delle interazioni rileva gli input dell'utente e modifica istantaneamente il mondo virtuale.

Il 2016 è stato definito l'anno della realtà virtuale, l'hardware, seppure ancora costoso per gli standard dei consumatori, diventa più diffuso insieme a software come Unity o Unreal Engine 4 che forniscono un framework per lo sviluppo e la creazione di contenuti di ogni genere. Infatti nonostante comunemente la tecnologia VR viene associata al settore dell'intrattenimento, la realtà virtuale ha enormi potenzialità che vanno oltre la fruizione di contenuti ludici lato multimediale [3]. A tal proposito Nvidia ha sviluppato piattaforme hardware e software che vogliono dare una spinta all'utilizzo della realtà virtuale anche negli ambienti della produzione industriale e del commercio.

Nel settore Automotive la virtualizzazione e la simulazione 3D offrono un valore aggiunto sia in contesti ingegneristici che nella comunicazione di prodotto, ne sono un esempio le simulazioni di guida o la valutazione del modello CAD in scala reale.

In campo medico viene utilizzata per facilitare l'addestramento degli operatori sanitari che, immersi in una simulazione di un evento realistico, possono sperimentare più e più volte l'operazione di intervento ed allenare la propria capacità di adattamento a situazioni di stress.

Grande utilizzo della VR viene fatto anche nel mondo del training. Questo tipo di formazione consente all'utente di immergersi totalmente nell'esercizio di training in condizioni di assoluta sicurezza, così da apprendere nel migliore dei modi i metodi di lavoro e le tecniche di sicurezza.

Questa tecnologia quindi si adatta perfettamente a diverse esigenze perché rende semplice la simulazione, la pianificazione, l'interpretazione, la visualizzazione dei dati scientifici e in generale l'indagine di tutti quei fenomeni che si rivelano essere difficili da descrivere in quanto complicati da percepire e misurare. Per tale motivo la VR trova grande impiego anche nel campo dell'astronomia, viene adoperata sia per esaminare nel dettaglio i risultati di complesse simulazioni dell'evoluzione degli oggetti astronomici che per puro scopo di rappresentazione con finalità didattica.

1.1 La NASA

Oggi gli scienziati della NASA³ sfruttano il potenziale delle realtà alterate per l'ingegneria spaziale e la scienza. Il loro obiettivo è quello di creare un ambiente collaborativo in cui gli utenti possano progettare, assemblare e interagire con i veicoli spaziali utilizzando strumenti standard e strumenti virtuali.

³ La NASA è responsabile del programma spaziale civile, nonché dell'aeronautica e della ricerca aerospaziale.

Come ha dichiarato lo stesso capo tecnologo del Goddard Space Flight Center (GSFC) Hughes: “[...] Per noi, la scoperta scientifica è una delle ragioni più convincenti per sviluppare una capacità AR/VR” [4]

Diversi sono i progetti in fase di sviluppo [5].

- Simulazione 3D della camera del vuoto termico: aiuta gli ingegneri a determinare se tutti i componenti dei veicoli spaziali si adatterebbero all'interno della struttura prima dell'inizio dei test.
- Servizio di assistenza robotica in orbita: combinando la vista della telecamera con i dati di telemetria si vuole creare un meccanismo che notifichi la possibilità di potenziali problemi prima che si verifichino.
- Analisi scientifica: utilizzando mappe digitali di elevazione e dati Lidar⁴ si vuole creare una simulazione 3D dei flussi e dei tubi di lava terrestri così che gli scienziati possano confrontare i dati raccolti in remoto con ciò che si può osservare sul campo.
- Visualizzazione tridimensionale dello spazio attorno al Sole per pianificare le missioni.
- Capire come si è formato il quartiere stellare locale: gli scienziati della NASA stanno ridefinendo la nostra galassia e il nostro modo di comprenderla. Dopo aver studiato gli stessi gruppi di stelle in sei dimensioni utilizzando materiale cartaceo, gli astronomi sono giunti a conclusioni diverse. Utilizzando la VR, che ha permesso di animare le stelle, gli astronomi hanno individuato stelle che potrebbero essere state classificate in gruppi sbagliati e che potrebbero appartenere a gruppi più grandi.

1.1.1 Chandra X-ray Center: la missione principale della NASA per l'astronomia a raggi X

Lo studio astronomico spesso è limitato alla proiezione bidimensionale di ciò che è possibile vedere nel cielo, di conseguenza si vorrebbero dei metodi più pratici e realistici che permettano di analizzare gli oggetti nello spazio. Tutto questo viene reso possibile dalla tecnologia VR combinata con ricchi set di dati.

Grazie alla collaborazione con la Brown University gli scienziati della NASA hanno aggregato i dati di Chandra, lo Splitter Space Telescope e le strutture terrestri per costruire un modello 3D del residuo Cassiopea A. Il programma VR/AR offre all'utente la possibilità di esplorare il residuo della supernova selezionando le parti con cui desidera interagire e su cui vuole ottenere informazioni aggiuntive [6].

Per visualizzare i dati 3D si sono sfruttati l'effetto Doppler e 3D Slicer, un programma che, modificato a seguito di un progetto di medicina astronomica ad Harvard, ha permesso di visualizzare e manipolare il modello 3D.

Premesso che Cassiopea A è il risultato di un'esplosione e che pertanto i suoi detriti stellari tendono ad espandersi radialmente verso l'esterno dal centro di esplosione è bastato utilizzare una geometria semplice per costruire il modello 3D. Dalla rappresentazione si è notato che esistono due componenti principali in questo residuo di supernova: un componente sferico probabilmente composto da elio e carbonio che si trova nelle parti esterne, e un componente

⁴ I dati Lidar sono dati ottenuti da un metodo di rilevamento, detto appunto Lidar, che misura la distanza di un oggetto illuminandolo con una luce laser e misurando poi la luce riflessa.

appiattito costituito dagli stati interni della stella situato nella regione interna e composto da elementi più pesanti quali ossigeno, neon, silicio e ferro.

Grazie a questa visualizzazione 3D gli astronomi sono stati in grado di costruire un modello di esplosione di supernova che permette di mappare posizione e orientamento dei getti di materiale che si verificano a seguito dell'esplosione (figura 1.1).

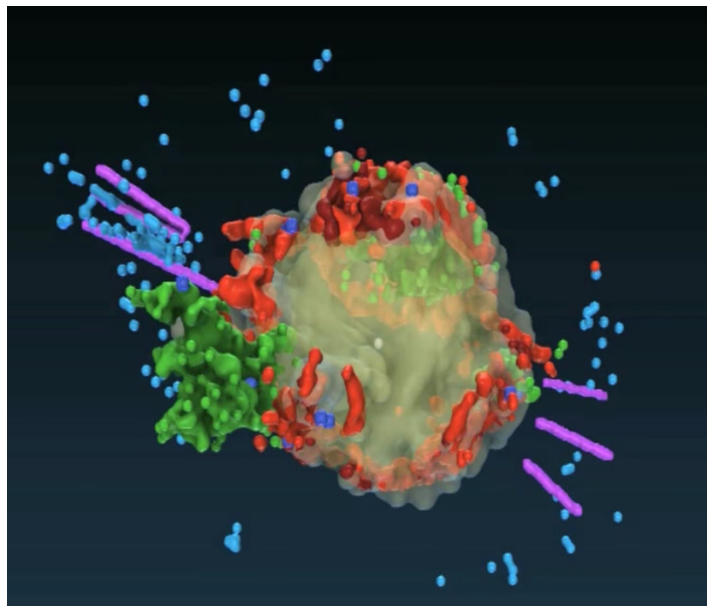


Figura 1.1 - Creazione della prima stampa 3D in assoluto di un residuo di supernova basato sui dati osservati.

1.2 Educazione astronomica e realtà virtuale

L'astronomia è una parte essenziale dell'educazione astronomica tuttavia alcuni studi hanno dimostrato come, nonostante la conoscenza aumenti con l'età, molto spesso gli studenti abbiano difficoltà a comprendere alcuni di questi eventi naturali con la conseguenza che pervengono ad una errata comprensione di ciò che è oggetto di studio in questa materia.

Secondo il cono della teoria dell'esperienza (Dale, 1946) gli studenti ricordano solo il 10% di ciò che leggono mentre ricordano il 90% di ciò che vivono, dunque i materiali tradizionali in uso nelle scuole non sempre sono sufficienti, i libri di testo spesso si rivelano essere inadeguati e i diagrammi 2D che rappresentano lo spazio 3D sono di difficile interpretazione. Ecco allora che il ricorso alla realtà virtuale può rappresentare un modo per supportare gli studenti nell'apprendimento e nell'indagine collaborativa. Con la VR gli studenti vengono immersi nell'universo e viene data loro la possibilità di vivere un'esperienza in prima persona così da poter osservare fatti che altrimenti sarebbero complicati da vedere [7].

La tecnologia VR permette di osservare un fenomeno da diversi punti di vista, rende possibili l'esplorazione, l'interazione, il senso della scala, la simulazione, la visualizzazione, facilita le attività di apprendimento alla scoperta e in particolare alla scoperta guidata⁵.

I sistemi VR desktop rappresentano un mezzo economico e disponibile utilizzabile per questi scopi. In base alle loro caratteristiche possono essere suddivisi in tre categorie:

⁵ Secondo Ormrod l'apprendimento alla scoperta è "un approccio all'istruzione attraverso il quale gli studenti interagiscono con il loro ambiente – esplorando e manipolando oggetti, ponendosi domande ed eseguendo esperimenti".

- Comunicazione interpersonale: aiutano l'apprendimento collaborativo permettendo la comunicazione e l'interazione tra diversi utenti.
- Navigazione delle informazioni: consentono agli utenti di navigare nel mondo virtuale e ottenere informazioni.
- Esperienza pratica: garantisce agli utenti l'interazione con gli oggetti del mondo; vengono solitamente applicati al campo della simulazione e della sperimentazione.

La tabella 1.1 mostra alcuni sistemi desktop VR utilizzati in ambito educativo.

NOME DEL SISTEMA	TIPO	METODO DIDATTICO	TARGET
Active Worlds	I	L'utente crea il mondo che desidera e il sistema fornisce funzioni di interazione sociale	Studenti delle scuole superiori e universitari, adulti
Virtual European School (VES)	I	Una comunità 3D fornisce materiale didattico, comunicazione e interazione multiutente	Studenti delle scuole medie
Virtual Museums, MoMA, Metropolitan Museum	II	Tour virtuali a 360 ° di sale espositive e musei	Tutti
Physics Education Research (PER)	III	Fornire ambienti VR per esaminare leggi fisiche come il movimento del rivestimento, il movimento circolare e le collisioni.	Studenti delle scuole superiori
Virtual Reality Physics Simulation (VRPS)	III	Fornire ambienti VR per esplorare fenomeni scientifici come propagazione delle onde, ottica a raggi, velocità relativa, macchine elettriche, ecc.	Studenti delle scuole superiori e universitari
Virtual Radioactivity Laboratory	III	Fornisce ambiente VR per analizzare la radioattività	Studenti universitari
Web Talk-I & II	I, II	Un avatar automatizzato guida gli utenti a visualizzare le mostre in tempo reale. Gli utenti possono discutere con l'esperto e altri utenti durante i tour virtuali	Tutti
Shrine Education Experience (SEE)	I, II, III	Sviluppa un museo virtuale, permette socializzazione e manipolazione nel mondo virtuale	Studenti delle scuole superiori

Tabella 1.1 - Alcuni sistemi VR desktop utilizzati in ambito educativo.

**Tipo I: comunicazione interpersonale, Tipo II: navigazione delle informazione, Tipo III: esperienza pratica.*

1.3 Applicazioni per esplorare l'universo

Negli ultimi anni si stanno sviluppando diverse applicazioni di realtà virtuale che hanno come obiettivo l'esplorazione interattiva dell'universo. Qui vengono descritte due tra le più importanti.

Space Engine (figura 1.2) sviluppata dall'astronomo e programmatore russo Vladimir Romanyuk. Questa applicazione è un simulatore spaziale monoutente che, tramite generazione procedurale, genera sistemi planetari, satelliti, asteroidi, ammassi stellari, nebulose e galassie. Per rappresentare l'intero universo in modo conforme a quanto affermato dalla scienza moderna, vengono utilizzati dati astronomici reali combinati algoritmi scientifici di generazione procedurale. L'utente può modificare la velocità del tempo e viaggiare attraverso lo spazio così da osservare tutti i fenomeni celesti che desidera. Gli oggetti astronomici sono caratterizzati da specifiche proprietà quali temperatura, massa, raggio, etc.. che vengono rese disponibili all'utente sulla barra di stato e in una finestra di informazioni.

Il software è scritto in C++, il motore utilizza OpenGL come API grafica e gli shader sono scritti in GLSL; è disponibile solo per PC Windows [8].

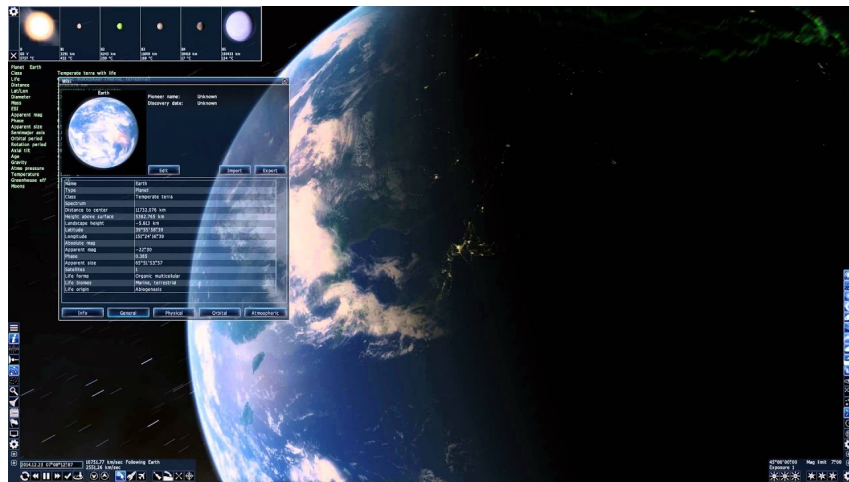


Figura 1.2 - Esempio di Space Engine.

Celestia (figura 1.3) è un programma software di astronomia 3D gratuito e open source rilasciato sotto licenza GNU General Public License sviluppato da Chris Laurel. Questo programma è disponibile per AmigaOS 4, Linux, macOS e Windows. Mostra e permette di interagire con gli oggetti astronomici tra cui le 118322 stelle del catalogo Hipparcos che si trovano ad una distanza massima di qualche migliaio di anni luce dal Sole. Esso, grazie alla teoria VSOP87 delle orbite planetarie permette di osservare eclissi lunari, eclissi solari, percorsi orbitali, pianeti, lune, asteroidi, comete, satelliti artificiali e veicoli spaziali, senza però simulare la gravità. Inoltre il movimento stellare non viene simulato e le galassie rimangono in posizioni fisse nonostante gli oggetti che fanno parte di un sistema planetario si muovano e le stelle ruotino attorno ai loro assi e si orbitino a vicenda in più sistemi stellari.

L'utente può viaggiare in tutto l'universo ad una qualsiasi velocità in qualunque istante del passato o del futuro, può cambiare il suo punto di vista e può dividere la finestra in più riquadri così da riuscire ad osservare più corpi celesti contemporaneamente. Il rendering cerca di far coincidere la visione umana nella posizione dell'osservatore nel modo più accurato possibile, sono adoperate mappe a falsi colori e nebulose multicolori; Gli artefatti della fotocamera come il riflesso e il riflesso dell'obiettivo non vengono riprodotti. Celestia è un software utilizzato

dalla NASA e dall'ESA nei loro programmi educativi e di sensibilizzazione e come base per il software di analisi della traiettoria [9].

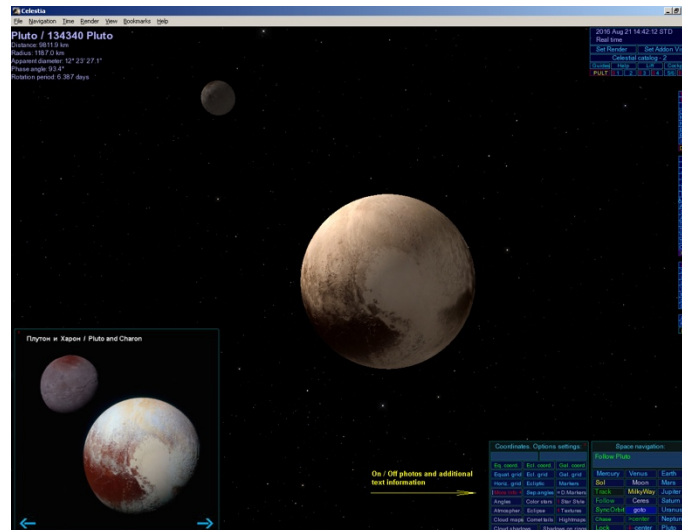


Figura 1.3 - Esempio di Celestia.

1.4 Altec: Astra Data Navigator

Astra Data Navigator è un'applicazione realistica e semplice realizzata da Altec. Essa ha come obiettivo la visualizzazione dei dati ottenuti dai satelliti roaming come Hipparcos al fine di dedurre nuove informazioni e caratteristiche dei corpi celesti basandosi semplicemente sulla posizione e alcune caratteristiche come raggio della stella rispetto a quello del Sole, intensità della luminosità (apparente) e classificazione spettrale

Dal momento che la scena è popolata da oggetti relativamente piccoli molto distanti tra di loro, affinché l'utente possa esplorare al meglio lo spazio egli viene inizialmente ancorato all'oggetto più vicino. Successivamente può scegliere se rimanere ancorato all'oggetto e spostarsi sfericamente attorno alla superficie dell'oggetto stesso in modo tale da osservarlo da ogni angolo come una camera di terza persona il cui focus è l'oggetto ancorato, oppure può disabilitare l'ancoraggio e ruotare la vista rimanendo fermo così da cercare gli altri oggetti della scena. L'utente ha inoltre la possibilità di decidere se spostarsi e ancorarsi ad un nuovo oggetto e lo spostamento della camera è implementato tramite il teletrasporto: la posizione dell'osservatore viene immediatamente trasferita sulla posizione di destinazione.

Le informazioni relative all'oggetto a cui l'utente è ancorato sono rese disponibili tramite un pannello posizionato nella parte destra della scena.

1.5 La nuova versione di Astra Data Navigator

La nuova versione di Astra Data Navigator vuole estendere la prima versione ampliando il numero dei corpi celesti rappresentati e apportando delle modifiche dal punto di vista del funzionamento.

Con il nuovo software viene introdotto il sistema solare: i pianeti e i satelliti sono posizionati all'interno della scena con la posizione reale relativa alle ore 00:00:00.000 del giorno 20 Giugno 2019 e sono soggetti ai moti di rotazione e rivoluzione che l'utente può decidere se attivare o meno.

Rispetto alla versione precedente di Astra Data Navigator, adesso l'utente può navigare all'interno della scena muovendosi liberamente lungo tutte le direzioni e secondo scale di velocità differenti. Nel momento in cui l'utente sceglie la nuova stella o il nuovo pianeta a cui ancorarsi, lo spostamento della camera non avviene più mediante teletrasporto ma la camera si sposta lungo la direzione del nuovo punto di ancoraggio fermandosi ad una certa distanza da questo così che l'oggetto possa essere visto nella sua interezza.

Anche la GUI è stata modificata: sono state modificate le funzionalità di alcuni componenti e, in particolare, è stata aggiunta la possibilità di trascinare il pannello raffigurante le informazioni relative all'oggetto che si sta osservando così che l'utente possa posizionarlo in un punto che non interferisca con l'oggetto astronomico stesso.

Nella nuova versione dell'applicazione inoltre l'utente, tramite tastiera, può decidere se utilizzare il cursore del mouse classico – attivo di default o tramite il tasto '9' nella parte superiore della tastiera alfanumerica – o un cursore tridimensionale – attivabile tramite il tasto '0' nella parte superiore della tastiera alfanumerica.

L'applicazione è predisposta per essere utilizzata con un proiettore a stereoscopia attiva e l'interazione con gli oggetti della scena è gestita tramite mouse 2D e tastiera.

CAPITOLO 2

La Realtà Virtuale

2.1 Cos'è la realtà virtuale

La realtà virtuale è un concetto complesso, essa non è soltanto tecnologia, è un mezzo di comunicazione, un'esperienza e di conseguenza dare una definizione univoca non è semplice.

Si potrebbe partire da una definizione più filosofica secondo cui “La realtà virtuale permette di vivere in un mondo *verosimile* in cui si mescolano il magico e il *reale*”, fino ad arrivare a definizioni date da studiosi e ricercatori. Per citarne qualcuna possiamo ricordare la definizione data dal professore Gregor Burdea che nel suo libro scrive che: “[...] *la realtà virtuale è un'interfaccia uomo-computer di alto livello che comprende la simulazione real-time di un mondo realistico e le molteplici interazioni con gli oggetti di tale ambiente attraverso canali sensoriali multipli*” [10].

Di fatto dunque la realtà virtuale non è altro che una simulazione il cui la computer graphics è adoperata per creare un mondo che sia visivamente realistico.

I ricercatori W. Sherman e A. B. Craig definiscono quattro elementi essenziali per la realtà virtuale:

- Mondo virtuale
- Immersione
- Presenza
- Interazione

Mondo virtuale

Il mondo virtuale è un ambiente che include un insieme di oggetti governati da una serie di regole e relazioni. È necessario che il mondo virtuale sia interattivo e multimodale, ovvero deve rispondere (modificandosi) agli input dell'utente, la risposta deve essere istantanea e la percezione del mondo deve coinvolgere i diversi canali sensoriali dell'utente.

Immersione

L'immersione misura la percezione di essere fisicamente presente nel mondo virtuale, riguarda la sensazione di sentirsi non solo osservatori di un ambiente bensì partecipi.

Presenza

La presenza (*being there*) è definita come la “sensazione di appartenenza” all'ambiente virtuale, è una percezione mentale.

La presenza in un ambiente virtuale richiede:

- Che l'ambiente virtuale sia concepito e realizzato al fine di massimizzare questa sensazione;
- Che l'utente sia “predisposto” ad accettare la sensazione di presenza in un ambiente che non è quello reale.

Esistono tre fattori per garantire una massima presenza:

- Massima qualità delle informazioni sensoriali;
- Massima mobilità e controllo dei sensori, i dispositivi devono potersi muovere in uno spazio sufficientemente grande;
- Alto livello di interazione.

Interazione

L'interazione in applicazioni di realtà virtuale è un elemento fondamentale per trasmettere agli utenti un naturale senso di coinvolgimento. La difficoltà nella gestione dei meccanismi d'interazione risiede nel fatto che il cervello umano è molto sensibile a eventi e comportamenti che differiscono da quelli reali a cui è abituato, diventa necessario che l'ambiente reagisca in maniera plausibile agli input dell'utente.

In figura 2.1 si sottolinea come i fattori di interazione, immersione e presenza sono strettamente collegati tra di loro.

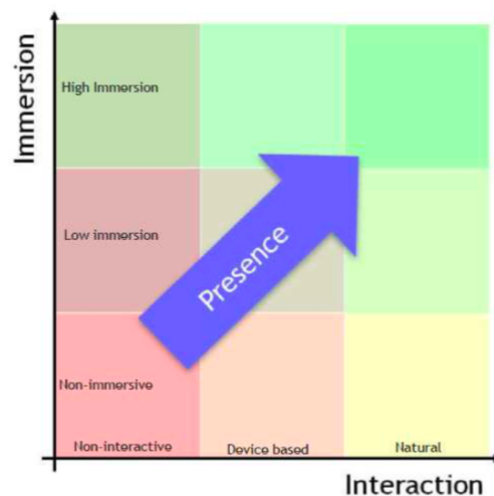


Figura 2.1 - Il senso di presenza aumenta all'aumentare del senso di immersione e interazione.

2.2 Tipi di Realtà Virtuale e concetti base

La realtà virtuale, a seconda del livello di immersione che offre, si suddivide in non immersiva e immersiva (figura 2.2).

- **Non immersiva:** l'ambiente digitale ha un minore impatto emotivo sull'utente che rimane consapevole della presenza del mondo esterno; solo una parte dei sensi vengono stimolati.

- **Immersiva:** l'utente viene completamente isolato dall'ambiente esterno e viene trasportato nel mondo virtuale. È la realtà virtuale propriamente detta dove tutte le componenti presenza, immersione e interazione sono presenti.



Figura 2.2 - A sinistra un esempio di realtà virtuale non immersiva, a destra un esempio di realtà virtuale immersiva.

La realtà virtuale dunque non è uno strumento che mira a ricostruire necessariamente la realtà, ma punta a “convincere” l'utente di essere nella realtà per permettergli di svolgere al meglio il compito che gli è stato assegnato, è un sistema complesso dove vi sono diversi concetti da considerare.

Di seguito verranno descritti i concetti più importanti.

Field of View

Il FOV (*Field of View* o *Field of Vision*) indica l'angolo sotteso dalla superficie visibile dal punto di vista dell'osservatore. Rappresenta l'estensione del mondo virtuale che viene visualizzato e viene misurato in gradi.

Il campo visivo di un singolo occhio è di 150° orizzontali e 120° verticali ma, poiché i campi visivi dei due occhi si sovrappongono parzialmente in orizzontale, l'area visibile (*campo visivo binoculare*) misura circa 180°-200°.

Per garantire una maggiore copertura del FOV umano è possibile aumentare la dimensione del canale visivo circondando completamente l'utente con delle pareti su cui vengono proiettate le immagini, oppure adoperando display che isolano la vista dell'utente dal mondo esterno.

Realismo

È necessario che il mondo virtuale sia in scala realistica.

Frame rate

Il frame rate rappresenta la frequenza con la quale vengono riprodotti i fotogrammi. Il numero di immagini riprodotte in un secondo viene identificato dall'acronimo FPS (*Frames Per Second*). Per una normale riproduzione video si ha un valore di circa 30 FPS, per applicazioni di realtà virtuale, invece, è necessario che il frame rate sia elevato per ottenere un risultato fluido: occorre avere un valore di almeno 60 FPS.

Latenza

La latenza indica un ritardo che intercorre tra un segnale in input e la rispettiva risposta in output. La latenza del sistema è data dalla combinazione della latenza di sensori di input, del sistema di tracking, della pipeline di rendering, del display e del sistema di sincronizzazione. Solitamente una latenza inferiore ai 100 ms non viene percepita direttamente, ma una scena

in movimento da l'impressione di essere instabile. Dunque affinché un'esperienza di realtà virtuale sia realistica e fluida, è necessario che la latenza sia inferiore ai 20 ms.

Suono

Il suono è molto importante al fine di aumentare il senso di immersione. Esso deve essere fisicamente spazializzato in modo tale che il cervello umano possa individuare la posizione di emissione del suono.

Per ricavare informazioni sulla direzione di emissione di un'onda sonora si sfruttano tre fattori principali:

- Il ritardo di ricezione del suono tra le due orecchie (*Interaural time Delay*).
- La differenza di volume e intensità del segnale per le attenuazioni della testa (*Interaural Intensity Delay*).
- Filtraggio del suono legato all'interazione acustica con tronco, testa e orecchio esterno.

Coerenza movimento - prospettiva

I dispositivi di tracking hanno il compito fondamentale di comprendere l'orientamento e la posizione dell'utente nel mondo virtuale. Devono essere dispositivi che supportano la mobilità e che garantiscono un'interazione il più naturale possibile, devono essere intuitivi, semplici da utilizzare e funzionali.

2.3 Stereoscopia

La visione stereoscopica è una caratteristica del sistema visivo umano. Avviene quando la parte del singolo campo visivo monoculare dell'occhio destro si sovrappone alla stessa parte visibile anche dal singolo campo visivo monoculare dell'occhio sinistro: le due visioni prospettiche bidimensionali si uniscono dando origine ad un'immagine tridimensionale, con associate le informazioni sulla profondità e sulle distanze.

La percezione della profondità può essere influenzata da:

- fattori psicologici come ombre, occlusioni, texture e prospettiva atmosferica;
- fattori fisiologici:
 - **Accomodamento:** viene modificata la lunghezza focale per avere l'immagine messa a fuoco sulla retina.
 - **Convergenza:** è la rotazione degli occhi per allineare l'immagine destra e l'immagine sinistra in una sola.
 - **Disparità binoculare:** dovuta alle visioni diverse dei due occhi.
 - **Parallasse di movimento:** è la differenza delle viste dovute ai movimenti della scena o di chi guarda; oggetti più vicini sembrano muoversi più velocemente di quelli distanti.
 - **Stereopsi:** il cervello unisce le due immagini ed elabora le informazioni di profondità.

La stereoscopia quindi è una tecnica che permette di trasmettere un'illusione di tridimensionalità, analoga a quella che il nostro cervello genera grazie alla visione binoculare. L'idea di base è quella di ricreare due diverse immagini corrispondenti a ciò che vedrebbe l'osservatore con l'occhio destro e sinistro.

Per la realizzazione di questo meccanismo si possono utilizzare sistemi non intrusivi (*autostereoscopici*) come i sistemi basati sulla parallasse, i display volumetrici e i display olografici oppure sistemi basati su "occhialini". Questi ultimi possono essere di due tipi.

- **Head mounted:** si utilizzano due microdisplay diversi posizionati davanti agli occhi, in tal modo vengono visualizzate due immagini destra e sinistra ognuna sullo schermo posizionato davanti al rispettivo occhio. È il caso dei più famosi HMD.
- **Multiplexed:** visione stereoscopica riprodotta su un'unica superficie di visualizzazione.

2.3.1 Sistemi multiplexed

Per i sistemi multiplexed, in cui lo schermo è unico, è necessario adoperare apposite tecniche affinché la sensazione di tridimensionalità risulti reale.

- Multiplexing temporale
 - Stereoscopia attiva
- Multiplexing spaziale
 - Stereoscopia passiva

Stereoscopia attiva

La stereoscopia attiva si basa sull'utilizzo di immagini sequenziate nel tempo. Il sistema di generazione dell'immagine (monitor o proiettore) è sincronizzato tramite un segnale ad infrarossi con l'elettronica di un paio di occhialini (*shutter glasses*) le cui lenti sono costituite da degli LCD che vengono resi opachi e trasparenti in modo alternato. Al fine di percepire l'immagine come continua è necessario che il tempo di refresh sia di almeno 50 immagini al secondo per un occhio, ovvero 100 Hz totali.

Stereoscopia passiva

La stereoscopia passiva permette la visualizzazione di immagini tridimensionali con occhiali che non sono dotati di alcuna tecnologia elettronica.

La due immagini vengono proiettate contemporaneamente sullo stesso schermo e la loro separazione può avvenire tramite tecniche basate su:

- Luce polarizzata
- Anaglifi
- Color multiplexing

Luce Polarizzata

La luce è un'onda elettromagnetica caratterizzata da ampiezza, lunghezza d'onda e fase che oscilla nello spazio lungo una direzione di propagazione; la polarizzazione descrive l'orientamento della sua oscillazione.

Per attuare tale tecnica si utilizza un doppio proiettore le cui lenti sono dotate di filtri polarizzatori che convertono la luce non polarizzata in una polarizzata facendo passare solo la componente della luce che oscilla nel piano definito dal polarizzatore (figura 2.3).

La luce può esser polarizzata in modi differenti:

- **Lineare** : i filtri sono orientati a 90° l'uno rispetto all'altro. Tale tecnica ha tuttavia il grande svantaggio che, inclinando la testa, si perde la percezione della tridimensionalità in quanto la luce non viene più filtrata correttamente.
- **Circolare** : la luce viene polarizzata a spirale di senso opposto per immagine destra e sinistra. In questo modo rotazioni o inclinazioni della testa non fanno perdere l'effetto stereoscopico.

Vantaggi:

- Non sono richieste costose schede grafiche
- Costi contenuti per l'hardware di proiezione

Svantaggi:

- Può essere usato solo in ambienti in cui sono richiesti pochi movimenti della testa (esempio al cinema)
- Lo schermo di proiezione deve essere di uno speciale materiale polarizzante

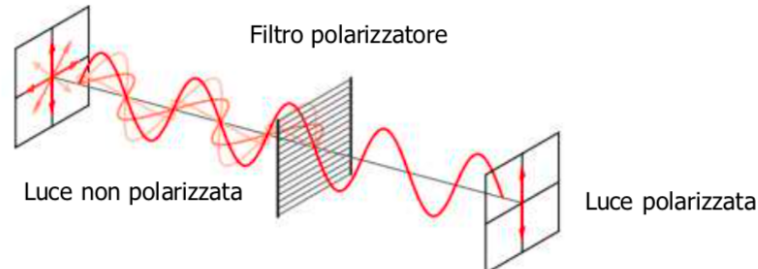


Figura 2.3 - Esempio di filtro polarizzatore.

Anaglifi

Gli anaglifi utilizzano un filtraggio nello spazio dei colori per la separazione delle immagini (figura 2.4).

Si utilizzano occhiali con lenti di colore cromaticamente opposti (solitamente rosso e ciano) e l'immagine viene ottenuta per sovrapposizione di due fotogrammi di uno stereogramma che subiscono un processo di colorazione distinto. In particolare si usa il rosso per l'immagine sinistra e il ciano per l'immagine di destra.

Vantaggi:

- Estremamente economica
- Facilmente fruibile

Svantaggi:

- La predominanza di rosso/ciano impatta sulla resa cromatica dell'immagine risultante

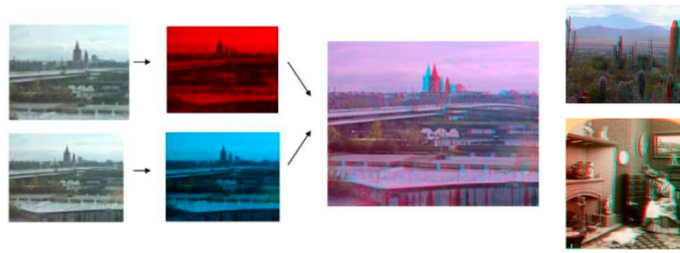


Figura 2.4 - Esempio di anaglifi.

Gamma di colore

Per la realizzazione della tecnica della gamma di colore viene impiegata una ruota di colori alternati integrata nel proiettore. Vengono utilizzate due basi diverse ($R_1G_1B_1$) per l'occhio destro e ($R_2G_2B_2$) per l'occhio sinistro e per ogni occhio vengono filtrate le componenti specifiche (figura 2.5).

Vantaggi:

- Scarsi fenomeni di ghosting
- Indipendenza dalla rotazione della testa
- Migliore separazione delle immagini
- Qualsiasi superficie può fungere da schermo

Svantaggi:

- Costi molto elevati
- Le immagini subiscono un'alterazione cromatica che va corretta via software

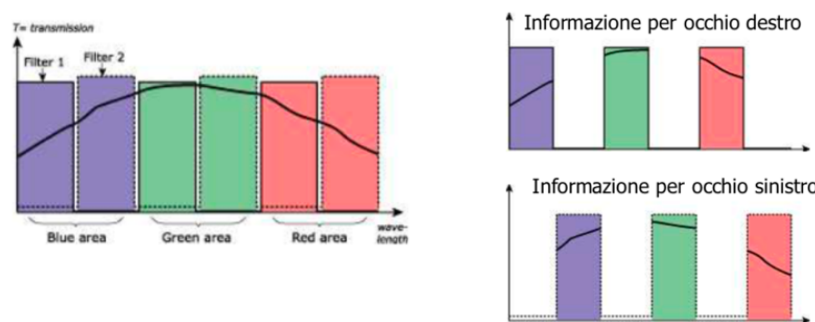


Figura 2.5 - Esempio di gamma di colore.

2.4 Realtà Virtuale: passato e presente

L'espressione "Realtà Virtuale" è stata coniata da Jaron Lanier nel 1989, anno in cui è stato fondato il *Virtual Programming Languages Research Lab*, ma già a partire dagli anni Cinquanta ci furono i primi tentativi per realizzare ambienti immersivi.

Nel 1955 il regista statunitense Morton Heilig presentò il concetto di "cinema d'esperienza", un cinema in grado di coinvolgere tutti i sensi, un cinema che immergeva lo spettatore nelle scene. Questa sua idea fu concretizzata nel 1962 con la realizzazione di *SENSORAMA*, un simulatore passivo di motociclo. La macchina era caratterizzata da uno schermo stereoscopico

attraverso cui venivano mostrate le immagini all'utente posizionato appositamente su una sedia girevole che si muoveva insieme alla simulazione. Per aumentare il senso di immersione venivano prodotte delle vibrazioni sulla sedia e sul manubrio e per simulare il vento si utilizzava un phone a diverse velocità.

Sensorama dunque utilizzava molti dei concetti della moderna VR come visione stereoscopica, feedback tattile, suono spazializzato, vento e odori, tuttavia mancavano l'interazione e l'utilizzo di immagini sintetiche.

Il primo vero visore di realtà virtuale nacque nel 1968 per opera dello scienziato e informatico Ivan Sutherland. Il dispositivo utilizzava due tubi a raggi catodici ed elementi ottici che proiettavano le immagini direttamente negli occhi dell'utente. Nonostante ciò questo prototipo presentava un difetto, era troppo grosso e pesante, e per poter essere utilizzato era necessario avere un'asta collegata al soffitto che sorreggesse il casco.

Nel 1982 Atari, azienda leader nel settore dei videogiochi, fondò un laboratorio dedicato alle ricerche sulle tecnologie di realtà virtuale che però venne chiuso a seguito della crisi del mondo dei videogiochi. Fu in quella circostanza che i due pionieri della realtà virtuale, Jaron Lanier e Thomas Zimmerman, crearono una nuova azienda: la VPL Research che sviluppò diversi dispositivi quali: Data Glove, Eye Phone e Audio Sphere.

Negli anni '90 iniziarono a diffondersi le prime disaffezioni verso la realtà virtuale e Jaron Lanier identifica tre cause principali:

- Erano necessarie elevate capacità di calcolo che solo le workstation di fascia alta potevano garantire;
- La mancanza di interfacce ad alta qualità, in particolar modo di display ad alta risoluzione e di strumenti di interazione adeguati, causava una user experience povera;
- La mancanza di software adeguati per la progettazione, sviluppo e gestione.

Negli ultimi 15 anni grazie ai miglioramenti delle GPU e alla disponibilità di software (anche Open Source) si è assistito alla rinascita della realtà virtuale. Sono aumentate le applicazioni destinate alla realtà virtuale e di conseguenza sono aumentate anche le tecnologie dedicate a questo scopo. HTC, Playstation VR e Oculus VR sono solo alcune delle aziende che hanno sviluppato visori sempre più all'avanguardia capaci di offrire performance elevate.

Nonostante ciò la realtà virtuale ancora oggi non è una tecnologia di massa e i motivi sono diversi.

- È più difficile ingannare i sensi: oggi gli utenti sono sempre più attenti e notano ogni piccola incoerenza con la realtà.
- Coinvolgimento non totale: sono stimolati maggiormente solo i sensi "principali" (vista e audio), gli altri vengono stimolati solo in parte o per niente.
- Spesso è monoutente: non è considerata come un'esperienza socialmente coinvolgente.
- Problemi di sicurezza: alcune interfacce tendono ad essere invasive e usi prolungati potrebbero provocare fatica/nausea.
- Costi dell'hardware: la realtà virtuale richiede componenti costosi e delicati.
- Componenti di interfaccia: HW richiede di essere migliorato in termini di risoluzione, complessità, velocità e dimensioni.

2.5 Limiti della realtà virtuale

Negli ultimi anni, nonostante i notevoli progressi, la realtà virtuale mostra ancora dei limiti legata all'utilizzo. I principali sono:

- **Costo:** la tecnologia VR richiede l'utilizzo di strumenti che spesso si rivelano non essere alla portata di tutti a causa del loro costo molto elevato.
- **Sicurezza:** spesso molte interfacce risultano essere invasive e, specialmente dopo un uso prolungato, causano una perdita della percezione di ciò che sta accadendo nella realtà.
- **Hardware:** l'hardware e in particolare i dispositivi di input possono essere ingombrati e d'intralcio in quanto potrebbero richiedere di essere collegati all'elaboratore tramite cavo.
- **Motion sickness:** l'utilizzo della realtà virtuale, specialmente quando l'applicazione chiede di muoversi nello spazio, può causare affaticamento visivo, mal di testa e disorientamento che spesso possono portare l'utente ad avere nausea. Questo sintomo può essere dovuto alla percezione visiva introdotta dall'auto-movimento, gli occhi dicono al cervello che ci si sta muovendo ma in realtà il corpo si sente come se fosse fermo. Diversi sono gli studi che hanno come obiettivo la risoluzione di questo problema, ad esempio nel 2017 il laboratorio di computational imaging di Stanford ha realizzato un prototipo di visore che aveva lo scopo di ridurre il senso di nausea in base ai parametri ottici.

2.6 La realtà virtuale nel settore aerospaziale

Il campo della realtà virtuale porta con sé molti vantaggi e trova ogni giorno sbocchi e declinazioni diverse, ad esempio nell'ambito dell'industria aerospaziale da la possibilità di adottare nuove modalità di sviluppo riducendone i costi. Permette di migliorare e velocizzare l'elaborazione dell'infrastruttura e delle operazioni dei sistemi spaziali e permette di accelerare il processo di apprendimento, specialmente nel momento in cui spesso i materiali di studio si rivelano essere limitati.

A partire dalle analisi di progettazione dei centri di controllo e dallo sviluppo di operazioni di attività extraveicolare (EVA) fino all'addestramento di missioni per l'equipaggio e i controlli di terra [11], la VR può fornire metodi vantaggiosi per programmare e condurre operazioni di volo spaziale. Permette di promuovere nuove tecniche e facilita la comunicazione in un ambiente di lavoro collaborativo affinché i diversi esperti (ingegneri, scienziati, ricercatori, etc.) possano collaborare al fine di raggiungere l'obiettivo comune. Viene utilizzata nella fase di progettazione e produzione per interagire con la geometria CAD così da rendere queste fasi più efficienti e soprattutto permette di garantire assistenza remota in caso sia necessario offrire servizi di manutenzione. Inoltre nelle fasi di addestramento garantisce che i praticanti riescano a vivere un'esperienza reale e sicura che permetta loro di migliorare e velocizzare il processo di acquisizione di conoscenza.

2.6.1 La realtà virtuale nello studio astronomico

Il più delle volte lo studio astronomico si basa su sistemi legacy⁶, database e materiali cartacei e di conseguenza può risultare complesso avere un quadro completo delle informazioni.

⁶ Un sistema legacy è un vecchio metodo, tecnologia, sistema informatico, o programma applicativo, "di, relativo a, o di essere un sistema informatico precedente o obsoleto," ma ancora in uso.

Dunque, nel momento in cui si vogliono approfondire degli aspetti più tecnici o provare ad elaborare nuove ipotesi, la realtà virtuale può essere considerata un ottimo mezzo di supporto per operare al meglio con questo tipo di dati [12]. A tal proposito una squadra della NASA guidata da Tom Grubb, ingegnere presso il Goddard Space Flight Center della NASA, ha realizzato un ambiente VR per studiare le caratteristiche di un quartiere stellare. Questo strumento *PointCloudsVR* (figura 2.6) è stato utilizzato per visualizzare e analizzare Gaia (osservatorio spaziale ESA) e il vento solare attorno a Marte e ha permesso di scoprire una nuova classificazione per un gruppo di stelle.

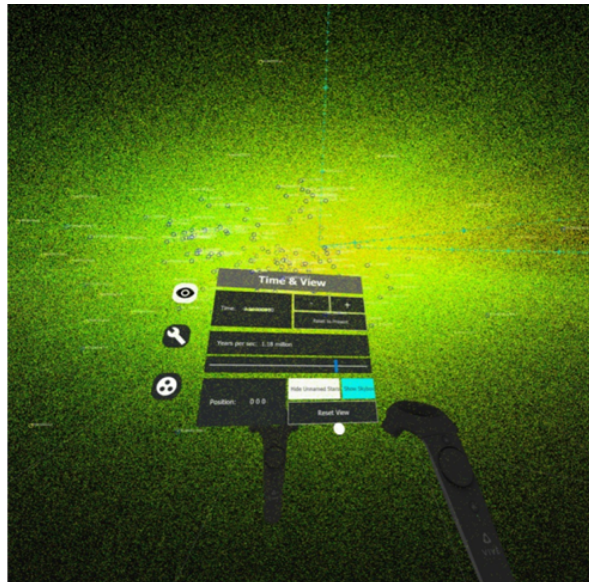


Figura 2.6 - Esempio di dati stellari di Gaia in *PointCloudsVR*.

Nonostante la naturalezza dell'interazione e il realismo delle rappresentazioni siano ancora aspetti che necessitano di essere migliorati così che si possa sfruttare al meglio questa tecnologia, oggi la realtà virtuale tende ad essere sempre più influente, permette di migliorare il processo di produzione, migliora gli strumenti di apprendimento e facilita le fasi di addestramento.

2.7 Realtà virtuale in Altec

Altec è un'azienda del settore aerospaziale che ha la responsabilità dei servizi di addestramento degli astronauti, di logistica e di supporto alle operazioni. Questa azienda ritiene che la realtà virtuale sia un ottimo strumento di supporto per programmi e progetti di ricerca e per migliorare l'analisi e lo sfruttamento dei dati basati sulle osservazioni spaziali.

Altec dispone di un VR-lab che prevede due possibili configurazioni.

VR Room (figura 2.7): un'apposita stanza è dedicata a questo scopo. Gli ambienti 3D e le simulazioni vengono visualizzate con la tecnica di stereoscopia attiva grazie ad un proiettore *Barco RLM-W14* che utilizza la tecnologia DLP (*Digital Light Processing*) per proiettare le immagini elaborate da un cluster di PC su un apposito PowerWall. Questo è un display di 1920x1200 pixel di risoluzione con frequenza di visualizzazione dei frame di 120Hz. La visualizzazione della scena avviene mediante occhiali stereoscopici attivi caratterizzati da schermi LCD e sincronizzati con il proiettore tramite infrarossi, mentre l'interazione con la

scena è possibile grazie ad un sistema di tracking⁷ e dei device di interazione. L'elaboratore è una macchina *Dell Precision Tower 7810* dotata di due processori *Xeon E5-2650 v3*, ognuno con 10 core e frequenza base 2.30 GHz, e di una scheda grafica *NVIDIA Quadro K5200*.



Figura 2.7 - VR Room di Altec.

Configurazione mobile (figura 2.8): è una versione semplificata della prima configurazione, permette di trasferire l'ambiente 3D ovunque occorre. È caratterizzata da un Tv-3D 4K, una workstation e un sistema di tracking portatile.



Figura 2.8 - Configurazione mobile del sistema di supporto alla VR in Altec.

⁷ Un sistema di tracking è un sistema che interagisce con l'unità di elaborazione del sistema trasmettendo l'orientamento del punto di vista dell'utente. Nei sistemi che permettono all'utente di muoversi trasmette anche informazioni circa la posizione, la direzione e la velocità di movimento.

CAPITOLO 3

Tecniche di realizzazione

All'interno di una scena Unity, quando la distanza tra la posizione della camera e quella di un oggetto è troppo elevata, ma si vuole che l'oggetto sia visibile, si va ad aumentare il piano far di clipping⁸ (figura 3.1), tuttavia questa soluzione porta alla generazione di ombre imprecise, di movimenti irregolari da parte degli oggetti che si trovano lontani dall'origine, e di artefatti di clipping e jitter⁹.

Inoltre lo z-buffer, che ha il compito di gestire le coordinate di profondità dell'immagine e quindi di decidere quali elementi della scena renderizzata saranno visibili, si imbatte in problemi di precisione.

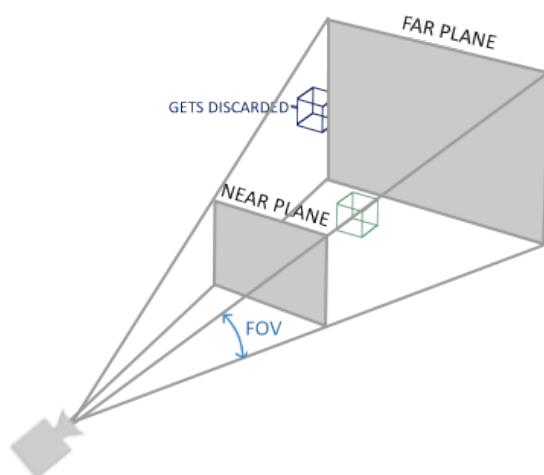


Figura 3.1 - Il frustum di visualizzazione è la regione dello spazio nel mondo modellato che può apparire sullo schermo. Gli oggetti più vicini alla telecamera rispetto al piano di clipping frontale o oltre il piano di clipping lontano non vengono disegnati.

Per rimediare alle problematiche sopra descritte esistono diversi approcci tradizionali:

- **Sistemi di coordinate locali multiple.**

Il mondo virtuale viene suddiviso in regioni più piccole ciascuna dotata di appositi meccanismi per essere gestita. Non appena l'osservatore supera la soglia di una specifica regione, il sistema di coordinate viene modificato e le nuove coordinate locali saranno legate alla nuova regione. (Grazie a questa tecnica le coordinate non arriveranno mai ad avere valori così grandi da causare jitter.)

⁸ Piano di far clipping: piano utilizzato per limitare l'output degli oggetti, definisce il limite massimo del volume di vista.

⁹ Jitter: sfarfallio o fluttuazione di un segnale di trasmissione o di visualizzazione dell'immagine. Nella computer grafica il termine jitter spaziale significa che un pixel ha posizione spostata di un certo delta rispetto alla sua posizione reale (così da ottenere un aspetto più naturale).

- **Shift delle coordinate a tratti in un mondo virtuale continuo.**

Il mondo virtuale non viene suddiviso ma viene gestito con un unico sistema di coordinate. Quando l'osservatore si sposta in una nuova area l'origine del mondo viene spostata nel nuovo punto di interesse: il mondo subisce dunque una trasformazione inversa e viene spostato a tratti mentre l'utente si sposta da un punto di vista all'altro. Viene sfruttata la tecnica del Floating Origin.

3.1 Virgola Mobile

In Unity la posizione di un `gameObject` è memorizzata in formato a virgola mobile a precisione singola utilizzando tre float (`Vector3`). Questo sistema funziona correttamente se si ha a che fare con una piccola area e quindi se si lavora con valori numerici piccoli, ma non può essere utilizzato quando diventa necessario avere un luogo esteso migliaia o milioni di chilometri.

Il degrado dell'accuratezza diventa un problema sempre più evidente man mano che ci si allontana dall'origine, la precisione della camera tende a diminuire e il `gameObject` lontano dall'origine non viene più posizionato in modo corretto, effetto evidente soprattutto durante le animazioni quando ci si sposta dall'origine.

Il problema è dovuto al metodo utilizzato dal computer per memorizzare la posizione X, Y, Z per il componente Transform utilizza lo standard IEEE per l'aritmetica in virgola mobile (IEEE 754) che si limita a 7 cifre decimali significative e di conseguenza più l'oggetto sarà lontano dall'origine più precisione in virgola mobile perde per rappresentare le sue coordinate.

Lo standard IEEE 754 a precisione singola è codificato con 32 bit, in particolare utilizza:

- 1 bit : bit di segno;
- 8 bit : larghezza esponente;
- 23 bit : mantissa, precisione significativa.

L'esponente può essere un numero intero sia con segno e nel caso assume valore compreso tra -128 e 127 sia un numero intero senza segno che assume valore da 0 a 255. Nel caso in cui si utilizza l'esponente con segno il valore 127 rappresenta lo zero effettivo, gli esponenti vanno da -126 a +127 e gli esponenti di -127 (tutti gli 0) e +128 (tutti gli 1) sono riservati per numeri speciali.

La mantissa indica il numero di cifre che portano significato. Questo standard comprende i 23 bit di frazione a destra del punto e un bit iniziale implicito con valore 1 a meno che l'esponente non sia memorizzato con tutti zeri. Dunque in realtà la precisione totale è di 24 bit (equivalente al $\log_{10}(2^{24}) \approx 7,225$ cifre decimali) [13].

Esempio (figura 3.2): il numero decimale $0,15625_{10}$ rappresentato in binario è $0,00101_2 = 1,01_2 \times 2^{-3}$. I tre campi nella rappresentazione IEEE 754 di questo numero sono:

- Segno = 0, perché il numero è positivo;
- esponente polarizzato = $-3 + 127 = 124$;
- frazione = $.01000..._2$.

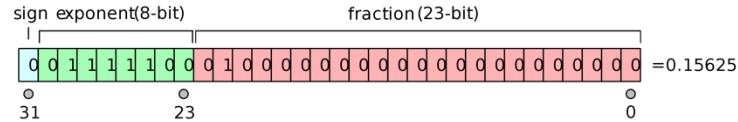


Figura 3.2 - Esempio rappresentazione di un numero secondo lo standard IEEE 754.

3.2 Floating Origin

Nel momento in cui l'osservatore si allontana dall'origine, specialmente se le sue coordinate sono memorizzate come virgola mobile, è probabile che si verifichino gli effetti di jitter spaziale e che le geometrie vicine modifichino la forma, l'aspetto e la posizione.

Il Floating Origin (FO) è considerato una soluzione generale e ottimale per tutto questo, come mostrato in figura 3.3, l'origine non rimane fissa rispetto al mondo ma viene fatta "galleggiare" con il punto di vista che sarà sempre centrato rispetto alla stessa origine.

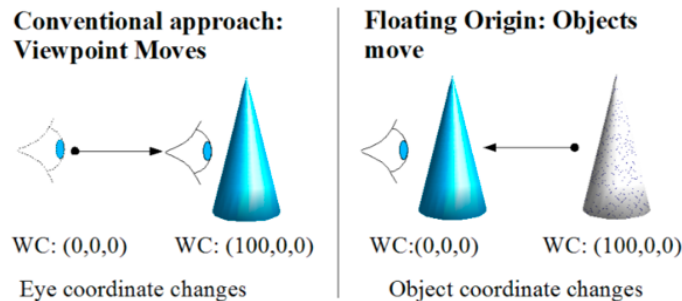


Figura 3.3 - Confronto tra il tradizionale movimento del punto di vista con il meccanismo del floating origin.

Tale tecnica sfrutta la regione di alta precisione in virgola mobile attorno all'origine dello spazio, ovvero la regione locale dove si richiede maggior accuratezza del rendering e del movimento, rendendo possibile l'eliminazione dello jitter spaziale, fornisce fedeltà di movimento costante e ottimale e consente ai dispositivi degli utenti finali di operare entro i limiti di precisione senza problemi.

Per realizzare il floating origin occorre effettuare la traslazione inversa del mondo ogni volta che il punto di vista è impostato su una nuova coordinata dell'applicazione o durante la navigazione libera. Non appena determinato il nuovo punto di vista, tramite una trasformazione di livello superiore si ottiene la trasformazione del mondo (WT) sull'intero insieme di oggetti che vengono spostati al contrario verso l'osservatore che rimane all'origine.

Normalmente un'applicazione tiene traccia della posizione degli oggetti in relazione tra loro e con il punto di vista dell'utente al fine di gestire le relazioni e le interazioni tra gli oggetti e rendere corrette le scene, tuttavia nell'approccio FO il tracciamento viene eseguito in modo diverso: le coordinate dell'oggetto nel sistema di visualizzazione sono uniche per la posizione dell'utente nel mondo perché sono sempre i valori più piccoli relativi alla posizione mondiale corrente dell'utente [14].

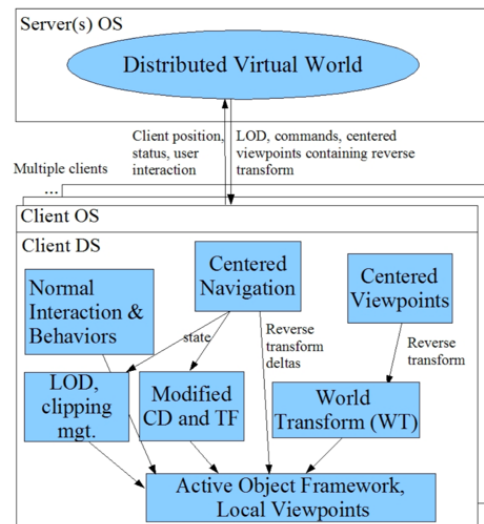


Figura 3.4 - Struttura del client in un sistema distribuito: mostra come il punto di vista centralizzato e i componenti di navigazione influenzano il WT che, a sua volta, trasforma gli oggetti. Una volta posizionati gli oggetti attivi, i delta di navigazione inversa inviati al framework degli oggetti attivi mantengono aggiornate le loro posizioni.

3.3 Scale Space

Congiuntamente al meccanismo di Floating Origin, anche la tecnica dello Scale Space rappresenta un'ottima soluzione ai problemi che si verificano quando si vogliono rappresentare spazi molto grandi. Tale tecnica permette di osservare oggetti che distano milioni di unità dalla camera e consente di esplorare tutto lo spazio superando anche i confini visivi, il tutto senza jitter.

L'ambiente viene suddiviso in più aree ognuna gestita con una propria scala e ad ogni area vi si assegna una camera che sarà centralizzata nell'origine e avrà un proprio frustum di visualizzazione (figura 3.5) [15].

Nel momento in cui la distanza di un oggetto dalla camera a cui è associato supera una certa soglia è l'oggetto ad essere spostato e/o ruotato, esso viene trasferito in un'altra area ridimensionata, viene dunque ridimensionato e riposizionato nella scena.

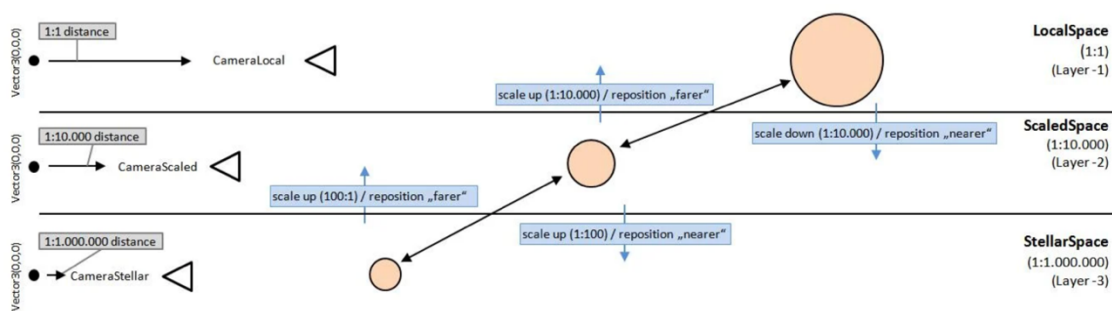


Figura 3.5 - Meccanismo scale space.

In particolare quando l'oggetto si avvicina all'origine di uno spazio viene inserito in uno spazio di dimensioni ridotte, quindi viene ridotto e riposizionato ulteriormente rispetto all'origine nello spazio reale; quando si sposta molto lontano dall'origine di uno spazio viene

inserito in uno spazio più ampio, quindi viene ingrandito e riposizionato più vicino all'origine nello spazio reale.

In questo modo quando l'osservatore si muove si muovono anche le camere con una velocità relativa alla scala dello spazio a cui appartengono.

Si ottengono camere a strati che riescono a superare i limiti di profondità dello z-buffer garantendo la piena visibilità dell'ambiente al momento della fase di rendering.

Per realizzare la tecnica dello Scale Space sono necessarie tre entità: Scale Space Object (SSO), Scale Space Camera (SSC) and Scale Space Manager (SSM).

3.3.1 Scale Space Object

Lo Scale Space Object gestisce il singolo oggetto all'interno della scena. Esso controlla se la posizione dell'oggetto ha raggiunto la soglia dell'area in cui si trova e in tal caso determina il nuovo spazio a cui dovrà appartenere. Sposta dunque l'oggetto aggiornando il riferimento della camera, lo aggiunge all'elenco di oggetti relativi alla nuova camera e trasferisce il Floating Origin della nuova camera.

3.3.2 Scale Space Camera

Lo Scale Space Camera gestisce una camera in uno specifico livello di spazio. Ogni singola camera esegue il rendering solo degli oggetti appartenenti al suo stesso layer e quindi è necessario impostare la profondità della camera per mantenere il rendering degli oggetti appartenenti agli spazi con le scale più piccole sopra quelle con una scala più grande. In questo modo, anche se nella scena un oggetto proveniente da uno spazio in scala più grande è effettivamente posizionato di fronte a un oggetto da uno spazio in scala minore, esso verrà comunque disegnato dietro mantenendo l'illusione della profondità.

Quando viene traslata, la camera verifica se la sua distanza dall'origine ha superato la soglia di spostamento e nel caso in cui accada la camera viene riposizionata all'origine della scena e gli oggetti appartenenti allo stesso livello di conseguenza vengono spostati della stessa distanza.

3.3.3 Scale Space Manager

Lo Scale Space Manager gestisce l'intero sistema mantenendolo sempre aggiornato: imposta i parametri globali e sposta correttamente le camere a seguito di un movimento dell'osservatore.

3.4 Asset Space Graphics Toolkit

La rappresentazione di ambienti molto grandi genera problemi oggi abbastanza noti e per questo motivo esistono diversi studi che hanno portato all'elaborazione di svariate tecniche che hanno come obiettivo quello di porre rimedio a problematiche di questo tipo.

In particolar modo per l'ambiente Unity l'asset in versione gratuita Space Graphics Toolkit [16] risolve le difficoltà analizzate sopra utilizzando il sistema di origine mobile. L'asset SGT tiene tutto vicino alla camera e popola la scena con modelli di oggetti adatti e scalati in modo opportuno a seconda della distanza dall'origine a cui si trovano.

La scena viene internamente suddivisa in celle e, utilizzando una coppia double - long di coordinate, per ogni oggetto si memorizza la cella corrente in cui questo si trova e la sua posizione in relazione alla cella stessa.

Le posizioni locali sono utilizzate per specificare la posizione di alta precisione dell'oggetto e indicano la sua posizione all'interno della cella in cui si trova, mentre le posizioni globali indicano la cella della griglia e dipendono dalle coordinate di alta precisione dell'oggetto stesso.

Non appena l'osservatore si muove gli oggetti vengono riposizionati e le coordinate globali vengono modificate se lo spostamento dell'osservatore supera la cella e quindi se le coordinate locali dell'oggetto superano il valore massimo raggiungibile.

Per eseguire correttamente questa soluzione occorrono entità specifiche che vanno a gestire in modo opportuno i vari componenti della scena.

3.4.1 SgtFloatingPoint

Memorizza la posizione del gameObject adoperando un'alternativa di altissima precisione ai valori di Vector3 per identificare un singolo punto nel sistema di origine mobile, utilizza infatti le coordinate locali e le coordinate globali. Questo componente si avvale della classe *SgtPosition* che memorizza le coordinate nel sistema di origine mobile e offre dei meccanismi per manipolarli. Permette ad esempio di determinare la distanza tra due punti o ancora la direzione.

3.4.2 SgtFloatingCamera

Contrassegna l'attuale gameObject come una camera e non appena la sua posizione si allontana troppo dall'origine (0,0,0) viene riportata all'origine. Questo meccanismo, oltre a garantire buone prestazioni perché gli oggetti cambiano occasionalmente le loro posizioni, permette di mantenere una buona precisione di rendering grazie al fatto che i valori delle posizioni della camera saranno sempre vicini all'origine.

3.4.3 SgtFloatingOrigin

Mette in pratica la tecnica del floating origin.

Memorizza la posizione della camera nell'universo e se assegnato agli altri oggetti, insieme al componente *SgtFloatingPoint*, memorizza la nuova posizione dell'osservatore contrassegnando l'oggetto come punto principale del sistema di origine mobile.

Essendo la posizione degli oggetti dipendente dalla posizione dell'origine, quando si fa interagire *SgtFloatingObject* (trasforma un normale gameObject in uno che funziona con il sistema di origine mobile) con *SgtFloatingPoint*, che a sua volta interagisce con *SgtFloatingOrigin*, non appena l'origine si sposta e quindi si aggiorna l'oggetto viene spostato e viene riposizionato rispetto alla nuova origine.

3.4.4 SgtFloatingScaler

Tramite la tecnica del floating origin gli oggetti, pur conservando le loro posizioni originali, vengono mantenuti vicini alla camera ma per essere visualizzati correttamente devono essere ridimensionati. *SgtFloatingScaler* ridimensiona l'oggetto in base alla sua distanza dall'origine rendendolo così visibile anche da una distanza maggiore del solito.

Capitolo 4

Progettazione dell'applicazione

Inizialmente il progetto era costituito da due scene: (stelle del catalogo Hipparcos, Sistema Solare). Queste sono poi state unite in un'unica scena così da rendere l'esplorazione più realistica. L'utente grazie a questa modifica può navigare l'universo visualizzando e analizzando tutti gli oggetti astronomici.

4.1 Catalogo Hipparcos

In Unity le classi *Monobehaviour* hanno una funzione di aggiornamento chiamata su ciascun frame, tuttavia, per evitare problemi di prestazioni, poiché nella scena sono presenti molti oggetti da aggiornare, il ciclo di aggiornamento è definito separatamente da quello originale e viene avviato solo a seguito di alcuni cambiamenti nella scena.

All'avvio dell'applicazione il componente *FileLoader* legge il file del catalogo Hipparcos e per ogni tupla crea un oggetto *Star* che memorizza tutti i dati significativi relativi a una stella che verrà aggiunta alla lista gestita dallo *StarManager*.

Al termine della lettura del file viene lanciato *UpdateModel* che per ogni stella contenuta nella lista crea il modello opportuno. Viene calcolata la distanza della stella dall'origine e quindi dalla camera e la sua magnitudine assoluta per decidere quale prefab deve essere istanziato per la specifica stella. Si stabilisce così se la stella ha stato:

- **selected**: la stella è selezionata, è visibile il suo modello 3D;
- **particle**: la stella è molto lontana ma visibile;
- **selectable**: la stella è lontana ma è possibile interagire con essa e avvicinarsi.

A seguito di ogni spostamento della camera la scena si modifica dinamicamente simulando lo spostamento e viene chiamato l'aggiornamento di *UpdateModel* e di conseguenza delle stelle.

Il diagramma in figura 4.1 mostra in modo semplificato il processo di generazione della scena raffigurante le stelle del catalogo Hipparcos.

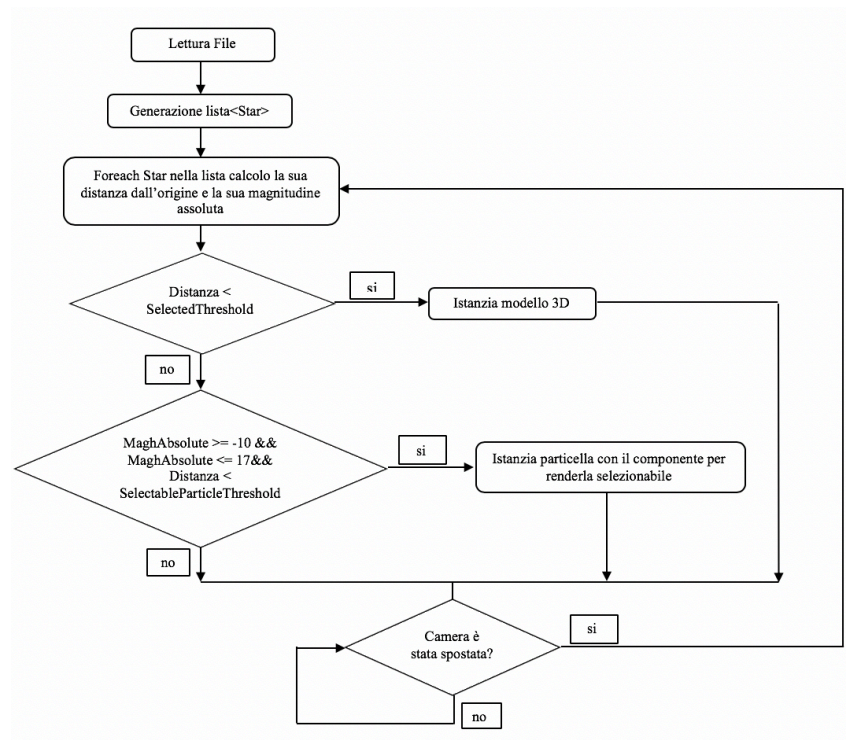


Figura 4.1 - Diagramma di flusso della generazione della scena.

4.1.1 Modelli

Ogni stella possiede delle caratteristiche (tabella 4.1) che determinano il rispettivo stato e conseguentemente il modello 3D corretto che serve per rappresentarla.

PARAMETRO	DESCRIZIONE
(X, Y, Z)	Coordinate della stella (in Parsec): (0, 0, 0) è la posizione del sole
RA(D), DEC(D)	Coordinate polari della stella (dalla terra)
MAGH	Intensità della luminosità (apparente)
BMV(J)	Frequenza di colore blu-viola
SP, CL	Classificazione spettrale
RSUN	Raggio della stella rispetto a quello del sole
DIS(PS)	Distanza della stella dal sole (in Parsec)
D(RSUN)	Distanza della stella dal sole (in raggio solare)

Tabella 4.1 – Caratteristiche di una stella.

4.1.1.2 Stella 3D

Per rappresentare la stella che l'utente sceglie di visualizzare da vicino si utilizza come mesh la sfera base Unity a cui, per aggiungere più realismo, è applicato uno shader che, dati alcuni parametri, genera e anima proceduralmente un materiale realistico.

Al prefab di tale gameObject è aggiunta anche una corona che essendo piatta e visibile solo frontalmente richiede di essere mantenuta sempre orientata verso la camera (per fare ciò è

stato associato lo script *Billboard*). Alla mesh della corona è assegnato un materiale che viene generato da un altro shader gestito, insieme allo shader assegnato alla sfera, dallo script *StarCreation* che utilizza una funzione per impostare la frequenza di colore BV, che verrà convertita in un colore dopo alcuni calcoli scientifici e quindi applicata al materiale. Nello specifico, data la frequenza di colore della stella si determinano le due coppie colore/temperatura più vicine a questa e, mediante interpolazione lineare, se ne calcola la temperatura espressa in Kelvin. Successivamente, dal valore della temperatura ottenuto, si ripristina il valore della frequenza di colore blu-viola che verrà utilizzato per approssimare il colore scientifico della stella. Queste approssimazioni vengono ottenute da una tabella di ricerca, non esiste infatti un'equazione veloce e precisa che permetta di calcolare ciò senza ricorrere alle equazione fisiche effettive.

4.1.1.3 Particelle

Il catalogo Hipparcos contiene centinaia di migliaia di elementi e dato che in ogni frame Unity scorre l'elenco di tutti questi oggetti, anche se non contengono alcun prefab al loro interno, risulta inefficiente creare un'istanza di tutte le stelle come gameObject.

Nel momento in cui l'utente decide quale stella osservare da vicino questa verrà rappresentata da un gameObject mentre tutte le altre, trovandosi ad una distanza molto elevata, possono essere ridotte a particelle.

Le particelle in genere sono billboard ovvero rettangoli che hanno una texture, spesso con canale alpha, che guardano sempre verso la camera. Queste non sono veri e propri gameObject ma sono solo degli effetti grafici, possono essere mosse nello spazio, possono avere una propria dimensione e un proprio colore.

Per definire e impostare i parametri di ogni singola particella si utilizza un sistema particellare (figura 4.2) gestito tramite script.

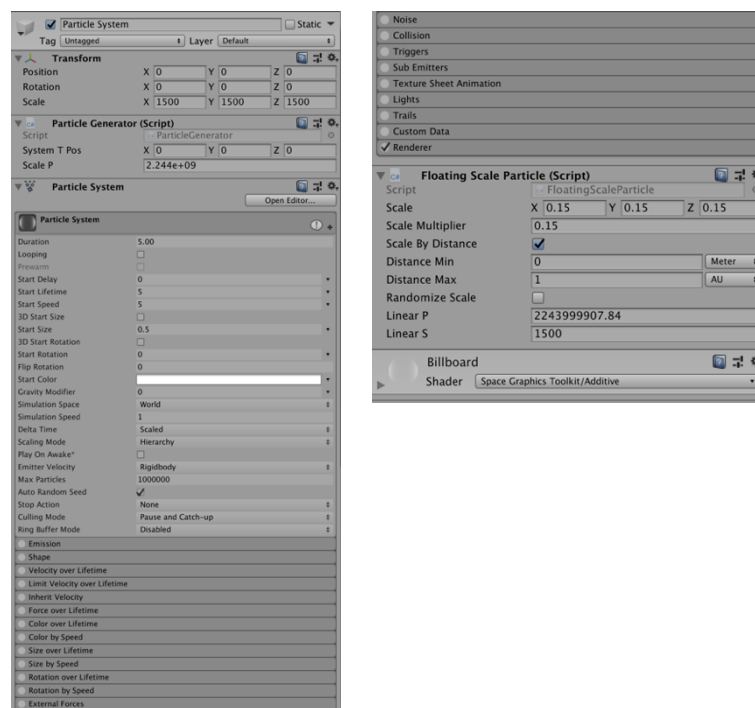


Figura 4.2 - Pannello del Sistema Particellare in Unity.

Volendo rappresentare tutte le stelle dell'universo, grande miliardi di anni luce, non appena il sistema particellare viene inizializzato deve essere ridimensionato rispetto alla sua distanza dall'origine (Appendice A.1), non risulta infatti sufficiente settare unicamente il valore *ParticleSystemScalingMode* che gestisce la scala di trasformazione applicata dal sistema in *Hierarchy*¹⁰.

Nel momento in cui si genera una particella questa viene aggiunta al sistema particellare responsabile e se ne settano le caratteristiche quali posizione, colore e dimensione (Appendice A.2).

Successivamente, molto importante è scalare la particella concordemente alla sua distanza dall'origine altrimenti questa non verrà visualizzata in modo opportuno all'interno della scena. Viene dapprima calcolata la differenza tra la posizione dell'attuale origine e quella della particella e a seconda del valore ottenuto si determina il fattore di scala per il quale dovrà essere moltiplicato il raggio della stella in questione (Appendice A.3).

Ogni volta che la camera viene spostata o si verifica un aggiornamento della scena tutte le particelle vengono cancellate e il sistema particellare viene ripopolato con le nuove particelle istanziate in modo opportuno.

La figura 4.3 mostra un esempio di stella vicina rappresentata dalla sfera base di Unity e più esempi di stelle lontane rappresentate da particelle.

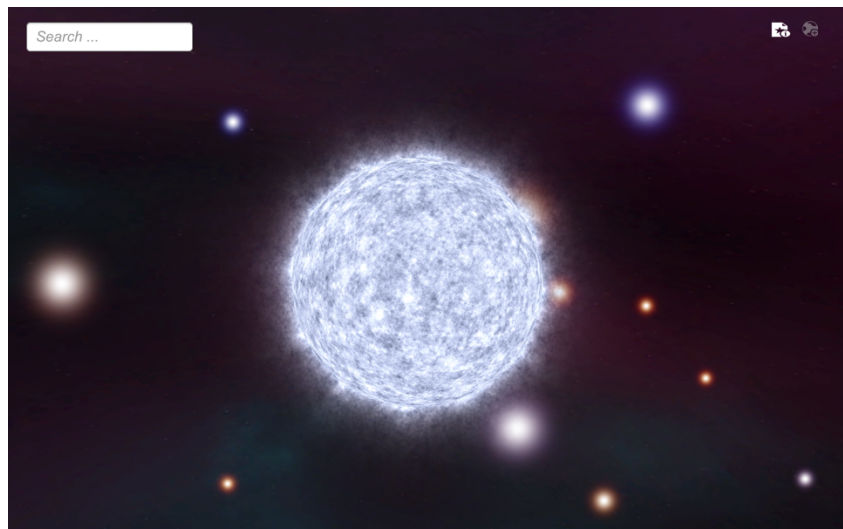


Figura 4.3 - Esempio dei modelli adoperati per le stelle: sfera e particelle.

4.1.2 Descrizione della stella

L'utente può scegliere se visualizzare o meno i dettagli della stella target tramite il bottone *InfoButton* che, non appena cliccato, apre un pannello contenente le informazioni.

Potrebbe capitare che alla sua apertura, l'elemento della UI si posizioni davanti alla stella e dunque potrebbe in qualche modo impedire la visione della stessa.

Pertanto oltre alla possibilità di nascondere, tramite il bottone *InfoStarButton* posto all'interno del pannello stesso, viene data all'utente la possibilità di spostarlo tramite trascinamento. Tale pannello, definito *draggable*, richiede di avere associato il componente

¹⁰ *ParticleSystemScalingMode = Hierarchy*: scala il sistema particellare utilizzando l'intera gerarchia di trasformazione, ereditando anche le modifiche di scalamento applicate sugli oggetti padre.

EventTrigger che una volta ricevuti gli eventi da *EventSystem* chiama le funzioni registrate per ciascun evento. Il pannello è controllato dallo script *StarInfoPanel* che per rendere possibile il comportamento di trascinamento, all'interno della funzione *Start*, come prima operazione, recuperare il componente *EventTrigger* e ne crea una nuova voce. Successivamente al trigger di eventi viene aggiunto l'evento desiderato, quindi l'evento di tipo Drag, in modo tale che la funzione *OnDragDelegate* venga chiamata non appena il sistema rileva il trascinamento. Questa funzione crea un raggio che parte dalla camera fino ad arrivare al *gameObject*, viene così calcolata la distanza che determinerà il trascinamento (Appendice A.4).

Per attuare questo comportamento occorre che l'elemento della UI abbia associato il componente *Event Trigger* e che la camera abbia associato il componente *Physics Raycaster* (figura 4.4).

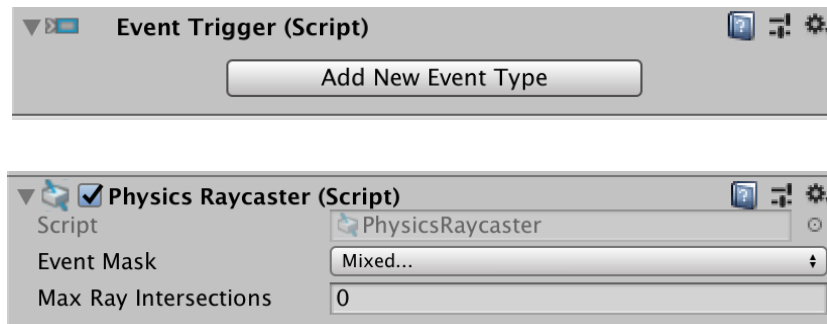


Figura 4.4 – Componenti da aggiungere al pannello e alla camera affinché il movimento di trascinamento funzioni correttamente.

4.1.3 Selezione Target (e spostamento)

Tutte le stelle che si trovano entro una certa distanza dall'origine vengono istanziate come particelle selezionabili e consentono all'utente di avvicinarsi ad esse.

Per compiere la navigazione verso la stella selezionata sono necessari diversi script che si occupano dapprima di individuare il target (stella verso cui l'utente vuole spostarsi) e successivamente di gestire lo spostamento della camera e l'aggiornamento delle altre stelle.

Vengono qui riportati gli script principali.

SgtFloatingTarget

Rende il *gameObject* a cui è associato un possibile target memorizzandone la sua posizione. Ciò permette di selezionare l'oggetto attraverso il componente *SgtWarpPin*.

SgtFloatingWarpPin

Permette di individuare il target selezionato dall'utente, confronta i valori della posizioni del punto della scena in cui l'utente ha cliccato con la posizione del target trasformata in posizione dello schermo ridimensionata. Se il punto dello schermo cliccato dall'utente coincide con la posizione di una stella selezionabile viene mostrata una finestra contenente il nome della stella e un bottone che se cliccato avvia lo spostamento della camera. Vengono inoltre abilitati e disabilitati i componenti della GUI che permettono di ricavare informazioni sull'oggetto.

SgtFloatingWarp

È la classe che gestisce tutti gli spostamenti. Individua il punto target e consente di avvicinarsi ad esso mantenendo una certa distanza in modo da non posizionare la nuova origine sulla stella ma davanti.

SgtFloatingWarpButton

Gestisce il funzionamento del bottone presente nella finestra contenente il nome della stella selezionata. Non appena il bottone viene cliccato, tramite la funzione *WarpTo* gestita dal componente *SgtFloatingWarp*, viene avviato lo spostamento verso il target.

SgtFloatingWarpSmoothstep

Con una velocità che sarà minore nella fase di partenza e arrivo, realizza lo spostamento vero e proprio verso il target. Al termine del movimento, tramite la funzione *transform.LookAt* ruota la camera in modo tale che questa sia rivolta verso la stella.

4.2 Sistema Solare

Il Sistema Solare è un sistema planetario costituito da diversi corpi celesti mantenuti in orbita dalla forza di gravità del Sole, ha un diametro di circa 240-260 AU ed è situato nel braccio di Orione della Via Lattea.

Esso è convenzionalmente suddiviso in due zone. Il sistema solare interno a cui appartengono il Sole e i quattro pianeti terrestri: Mercurio, Venere, Terra e Marte con i loro rispettivi satelliti, e il sistema solare esterno che comprende Giove, Saturno, Urano e Nettuno con i loro satelliti.

I corpi celesti che lo compongono sono caratterizzati da un moto di rivoluzione attorno al Sole che determina il loro spostamento all'interno del Sistema Solare stesso.

Per lo strumento AstraDataNavigator la posizione iniziale di ogni singolo corpo astronomico è stata calcolata tramite il tool *WebGeocalc*¹¹ scegliendo come riferimento temporale le ore 00:00:00.000 del giorno 20 Giugno 2019.

Il server *WebGeocalc* si basa sul codice Toolkit realizzato dal NAIF (Navigation and Ancillary Information Facility). Il sistema informativo presente alla radice prende il nome di “Spice” e i set di dati Spice primari vengono generalmente chiamati “Kernel”. Questi offrono dei meccanismi per calcolare le informazioni sulla geometria di osservazione così da facilitare la progettazione di missioni e la pianificazione di osservazioni scientifiche.

¹¹ <https://wgc.jpl.nasa.gov:8443/webgeocalc/#NewCalculation>

WebGeocalc offre capacità computazionali di Spice e fornisce un modo semplice e veloce per verificare se i dati di osservazione inclusi nell'archivio sono corretti. Ad esempio permette all'utente di ricavare i valori di distanza, velocità di rivoluzione e posizione che un corpo astronomico ha in un preciso istante di tempo rispetto ad un sistema di riferimento ben preciso.

4.2.1 Modelli

Per rappresentare i pianeti e i satelliti naturali si è utilizzata la sfera base di Unity a cui è stato assegnato un materiale dotato di texture opportuna per avere una rappresentazione realistica dell'oggetto, mentre per i satelliti artificiali come ISS e TGO si sono utilizzati modelli specifici. La figura 4.5 mostra la Terra insieme alla Luna, suo satellite naturale, e alla Stazione Spaziale Internazionale, suo satellite artificiale.

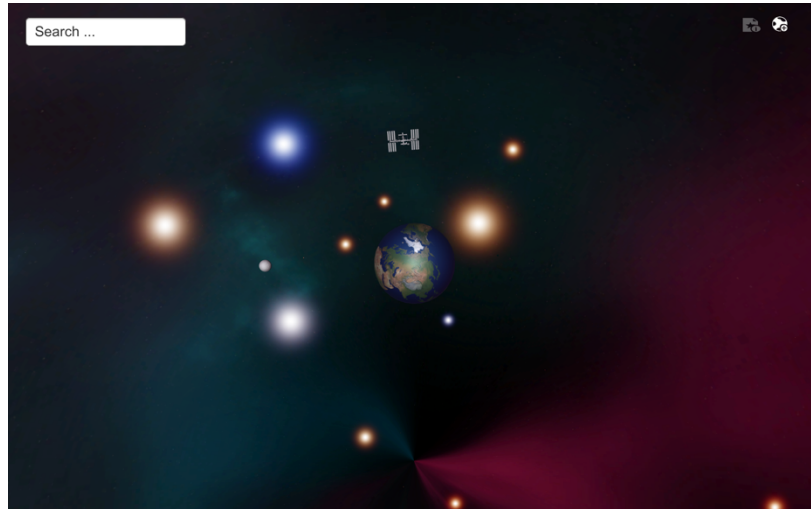


Figura 4.5 - Immagine raffigurante la Terra con i suoi satelliti.

Ciascun `gameObject`, affinché venga visualizzato con le giuste proporzioni, richiede di essere ridimensionato in modo opportuno secondo la sua posizione rispetto la camera. Per rendere possibile tale comportamento, ad ogni `gameObject` viene associato il componente *FloatingScalerPlanet* che monitora costantemente la distanza del pianeta dalla camera in modo tale che questo venga visualizzato correttamente quando l'utente si avvicina ad esso. Ad ogni frame, tramite la funzione `Update`, viene chiamata la funzione *SetDistance* (Appendice A.5) che calcola lo spazio tra il `gameObject` e la camera e in base al valore ottenuto determina il fattore di scala da applicarvi.

4.2.2 Il moto

Pianeti e satelliti sono caratterizzati da moto di rotazione e moto di rivoluzione. Per realizzare tali comportamenti si è sfruttata la funzione *transform.RotateAround* messa a disposizione da Unity che richiede come parametri la posizione attorno alla quale si vuole che avvenga la rotazione, l'asse di rotazione e la velocità di rotazione.

Il moto di rotazione è attivo di default al lancio dell'applicazione. Esso consiste nella rotazione dell'oggetto su se stesso e per questo motivo il primo parametro che dovrà essere passato alla funzione sarà la posizione dell'oggetto stesso.

Il moto di rivoluzione è gestito diversamente a seconda che l'oggetto sia un pianeta o un satellite. Per i pianeti questo moto non è attivo al lancio dell'applicazione ma sarà l'utente che, dopo aver selezionato come target un pianeta, deciderà tramite tastiera quando attivarlo ed eventualmente disattivarlo. La funzione *transform.RotateAround* associata al pianeta riceve come primo parametro la *transform.position* del sole poiché è attorno a questo che dovrà avvenire la rotazione. Inoltre per far sì che durante il moto la camera possa seguire il pianeta rivolgendosi verso di esso è necessario associare a questa le funzioni *Vector3.Lerp* e *transform.LookAt*. La prima riceve come parametri la posizione della camera stessa, la posizione del pianeta e la velocità con cui dovrà avvenire lo spostamento tra le posizioni

ricevute come primi due parametri, la seconda invece riceve come unico parametro la posizione del pianeta e ruota la camera in modo tale che questa lo guardi.

Per i satelliti invece la funzione *transform.RotateAround* che si occupa di gestire il moto è sempre associata all'oggetto ma riceve la *transform.position* del pianeta attorno cui il satellite dovrà ruotare. In questo caso il moto è attivo di default ma l'utente può comunque disattivarlo ed eventualmente riattivarlo in un secondo momento tramite tastiera.

Sia per il moto di rotazione che per il moto di rivoluzione, la velocità posseduta dagli oggetti è la velocità reale espressa in Mn/s¹².

4.2.3 Selezione Pianeta

L'utente ha la possibilità di avvicinarsi ad un pianeta e quindi di esplorarlo in due modi.

- **Tramite nome.**

Tramite la barra di ricerca è possibile trovare l'oggetto astronomico che si sta cercando. Nel momento in cui l'utente inserisce del testo, si verifica se quell'oggetto è presente nelle liste dei pianeti e dei satelliti istanziati e, se vi è corrispondenza, i risultati vengono mostrati sotto la barra di ricerca come pulsanti così che l'utente possa fare clic su di essi per spostarsi su di esso (figura 4.6).

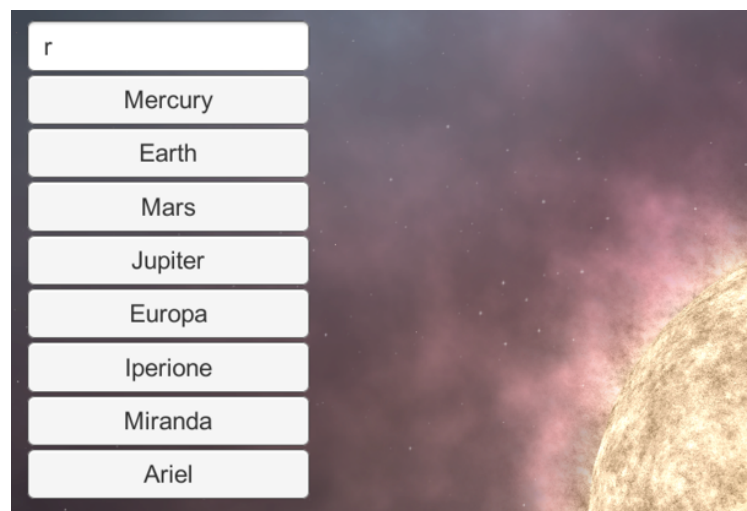


Figura 4.6 - Ricerca dell'oggetto astronomico tramite barra di ricerca.

- **Tramite esplorazione dello spazio.**

La camera non rimane ancorata ad una posizione della scena ma può essere traslata nello spazio tramite i tasti W-A-S-D. In questo modo l'utente può esplorare lo spazio che lo circonda scegliendo dunque se visualizzare più nel dettaglio un oggetto astronomico piuttosto che un altro.

4.2.4 Descrizione dell'oggetto astronomico

Ogni pianeta, così come ogni satellite, è accompagnato da una breve descrizione che mette in risalto le sue caratteristiche principali.

¹² Un megametro (Mn) è un'unità di misura SI di lunghezza ed equivale a 1000000 metri (m). Questa unità di misura viene solitamente utilizzata per misurare le dimensioni dei pianeti.

Nel momento in cui l'utente seleziona un pianeta o un stellite viene chiamata la funzione *ReadLine* del componente *FileTextInfo*. Questa riceve come parametro il nome del gameObject selezionato, legge un file contenente le informazioni e preleva quelle relative all'oggetto selezionato in modo tale da renderle disponibili all'utente tramite GUI. Il bottone *PlanetButton* permette di aprire/chiedere un'apposita finestra di testo rappresentante le informazioni.

4.3 Navigazione

L'utente può navigare all'interno della scena così da poterla esplorare nella sua interezza.

Per permettere all'utente di muoversi nella scena popolata da oggetti relativamente piccoli e molto distanti tra di loro, lo si ancora davanti al target da lui scelto e lo si fa navigare tra gli oggetti vicini.

L'utente può così avvicinarsi o allontanarsi e spostarsi a destra o a sinistra rispetto alla stella tramite i tasti W-A-S-D, o può spostarsi sfericamente ruotando la camera tramite mouse. Gli spostamenti tramite tastiera possono avvenire a diverse velocità, l'utente può scegliere se incrementarla o decrementarla tramite rotellina del mouse. La velocità inizialmente impostata al valore di 95 Mn/s aumenta in modo lineare fino al valore di 150 Mn/s dopo di che, se l'utente continua ad incrementarla assume i valori della velocità della luce, del doppio della velocità della luce, dell'unità astronomica, dell'anno luce e dei Parsec, sempre opportunamente espressi in Mn/s.

L'utente mentre esplora l'ambiente circostante può decidere di spostarsi ed ancorarsi ad un nuovo oggetto astronomico che diventerà così il nuovo target verso cui si dovrà spostare la camera e quindi la nuova origine del sistema.

4.4 GUI

Gli elementi dell'interfaccia utente necessitano di essere gestiti tramite Canvas ovvero una tela che racchiude e gestisce tali elementi.

Un Canvas ha tre diverse impostazioni di modalità di rendering:

- **Screen Space - Overlay:** gli elementi della UI vengono posizionati davanti allo schermo.
- **Screen Space – Camera:** il canvas viene posizionato ad una distanza predefinita dalla camera che ha il compito di renderizzarlo. Se la camera è impostata come prospettiva, come nel caso dell'applicazione Astra Data Navigator, allora il FOV della camera stessa produrrà una distorsione prospettica a cui gli elementi della GUI saranno soggetti.
- **World Space:** il canvas è trattato come un qualsiasi altro oggetto della scena, motivo per il quale è anche chiamata “interfaccia diegetica”.

Le applicazioni di realtà virtuale rendono complessa la visualizzazione della GUI poiché questa necessita di essere posizionata davanti alla camera in modo tale che gli elementi che la compongono vengano gestiti come oggetti della scena: viene utilizzata la seconda modalità di render.

Per controllare la scala e la densità dei pixel degli elementi dell'interfaccia si sfrutta invece il componente Canvas Scaler che, settato sulla modalità *Scale With Screen Size*, regola le

dimensioni di questi concordemente alle dimensioni dello schermo mantenendo le proporzioni di risoluzione con quella di riferimento.

4.4.1 Elementi di interazione

La GUI dell'applicazione sviluppata come si nota nella figura 4.7 si compone di due tipi principali di componenti di interazione:

- Pulsanti
- Campo di inserimento

Questa categoria di componenti ha il compito di gestire gli eventi dello spostamento o del tocco del mouse e dell'interazione tramite tastiera o controller. Sono componenti caratterizzati da almeno un *UnityEvent* che viene richiamato quando un utente interagisce con esso determinandone un cambiamento del suo stato (normale, selezionato, premuto, disabilitato). Per essere visibili, vengono combinati con dei componenti visivi. In questa applicazione si sono adoperate delle immagini che possiedono come trama una *sprite*¹³ per rendere visibili i pulsanti con cui l'utente deve interagire, e del testo per fornire informazioni aggiuntive sugli oggetti astronomici e sulla velocità di spostamento.

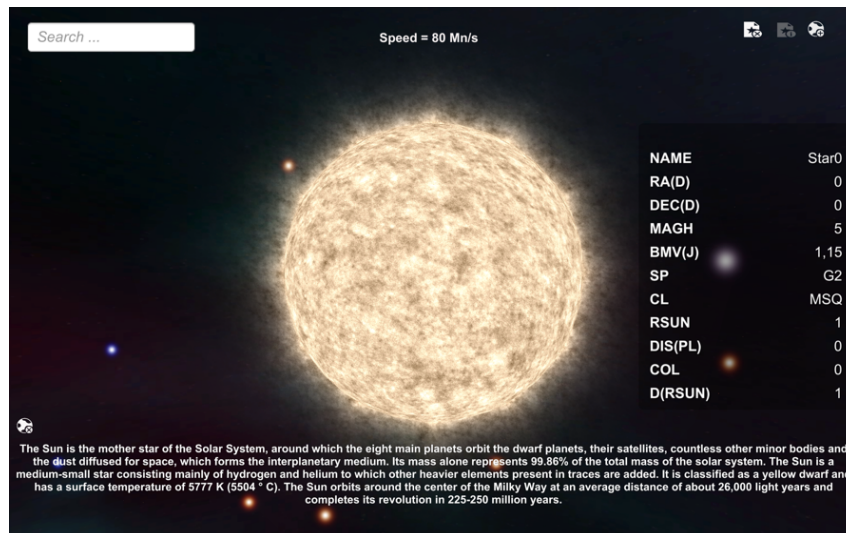


Figura 4.7 - Immagine della GUI di Astra Data Navigator.

4.5 Stereoscopia

Unity utilizza un set integrato di API grafiche selezionabili nell'editor. Per far funzionare correttamente la modalità stereoscopica, la scheda video deve riconoscere il flusso dati in output come flusso video stereoscopico. Occorre abilitare il supporto per le funzionalità specifiche per la realtà virtuale e caricare gli SDK della piattaforma di destinazione dell'applicazione, e successivamente scegliere la modalità di rendering per XR tra le quattro messe a disposizione da Unity: per questo lavoro si è scelta la modalità multi-pass.

Per elaborare un'immagine per l'occhio destro e un'immagine per l'occhio sinistro non occorre avere due camere differenti all'interno della scena ma basta impostare il parametro

¹³ Una *sprite* è un semplice oggetto 2D con immagini grafiche chiamate trame. Sono sempre rivolti verso la telecamera con un angolo perpendicolare tranne nel momento in cui vengono ruotati; in uno spazio 3D sono caratterizzati da assenza di spessore (coordinata Z).

Target Eye della camera in modo che le immagini riprese siano per entrambi gli occhi. Vengono inoltre impostati i due parametri di stereo separation e stereo convergenze che rappresentano rispettivamente la distanza tra gli occhi e la distanza a cui gli occhi convergono.

Capitolo 5

Criticità e Test

5.1 Criticità

Durante lo sviluppo del progetto è stato necessario affrontare e gestire alcuni problemi.

5.1.1 Caricamento e lettura del file del catalogo Hipparcos

Il catalogo stellare Hipparcos contiene informazioni su un numero elevato di stelle - circa 114000 - e dunque sarebbe opportuno che il sistema caricasse tale catalogo da database. Dove un database è in generale definito come un insieme di dati, organizzati in modo opportuno allo scopo, generalmente archiviati e accessibili tramite sistema informatico.

Nell'applicazione Astra Data Navigator i parametri di ciascuna stella vengono ottenuti e caricati tramite file. In generale la fase di lettura di un file di questo tipo avviene tramite l'utilizzo di diverse funzioni, nel caso di questo progetto si è adoperata la classe *stringReader* che tramite il metodo *ReadLine* consente di leggere una riga alla volta. La gestione dei dati di un file, richiede inoltre che, affinché la lettura avvenga in modo corretto, il file sia scritto con una formattazione idonea allo scopo e nota al programmatore. Ad esempio in questa applicazione, ogni riga del file, ciascuna relativa ad una stella, è suddivisa in 14 campi separati tra loro da un carattere di tabulazione che va tenuto in considerazione nel momento in cui occorre prelevare ogni singola proprietà e associarla alla giusta variabile.

La soluzione adottata in questo lavoro sicuramente non garantisce le massime prestazioni e per tal motivo attraverso dei test si è voluto analizzare il tempo impiegato per concludere la fase di lettura all'aumentare del numero di righe del file.

Per mezzo della funzione *Time.realtimeSinceStartup* resa disponibile da Unity si sono determinati il tempo intercorso tra l'inizio dell'esecuzione dell'applicazione e la lettura della prima riga del file, e il tempo in cui termina la lettura dell'ultima riga. Questi calcoli sono stati effettuati per 11 file differenti tra di loro per il numero di righe - si è prestata maggiore attenzione ai file aventi un numero di righe maggiore o uguale a 10000 - e per ogni file il test è stato ripetuto 50 volte così da avere una serie di dati da cui ricavare il tempo medio necessario alla lettura.

I test sono stati realizzati con due computer, si sono utilizzati un MacBook Pro sistema macOS High Sierra versione 10.13.1 con processore 2,8 GHz Intel Core i7, memoria 16 GB e scheda grafica Radeon Pro 555 2 GB - Intel HD Graphics 630 1536 MB (caso 1), e un computer Dell Precision Tower 7810 dotato di due processori Xeon E5-2650 v3, ognuno con 10 core e frequenza base 2.30 GHz, e di una scheda grafica NVIDIA Quadro K5200 (caso2).

Dopo un numero elevato di simulazioni, in entrambi i casi, si ottengono dei valori leggermente anomali rispetto alla crescita lineare che ci si aspetta. Questi sono probabilmente dovuti alle continue ripetizioni dei test che potrebbero causare un sovraccarico della memoria. Dunque è possibile concludere che, come supposto, la quantità di tempo necessaria per leggere il file aumenta in maniera lineare con l'aumentare del numero delle righe che si vogliono leggere.

Qui di seguito vengono rappresentati i grafici riguardanti i valori ottenuti durante i test.

5.1.1.1 Caso 1

Numero righe	Δt medio (s)
200	0,007
1000	0,153
10000	0,140
18000	0,184
22000	0,226
25000	0,365
50000	0,717
65000	0,686
100000	1,137
105000	1,175
113760	1,264

Tabella 5.1 - Intervallo di tempo medio ottenuto durante le simulazioni caso 1.

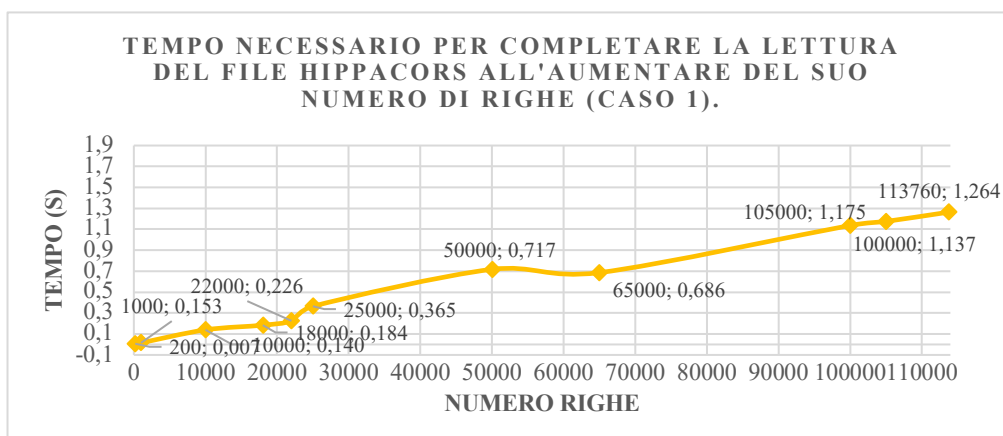


Figura 5.1 – Caricamento e lettura del file del catalogo Hipparcos caso 1.

5.1.1.2 Caso 2

Numero righe	Δt medio (s)
200	0,007
1000	0,063
10000	0,227
18000	0,356
22000	0,439
25000	0,349
50000	0,768
65000	0,926
100000	1,495
105000	1,582
113760	1,788

Tabella 5.2 - Caricamento e lettura del file del catalogo Hipparcos caso 2.

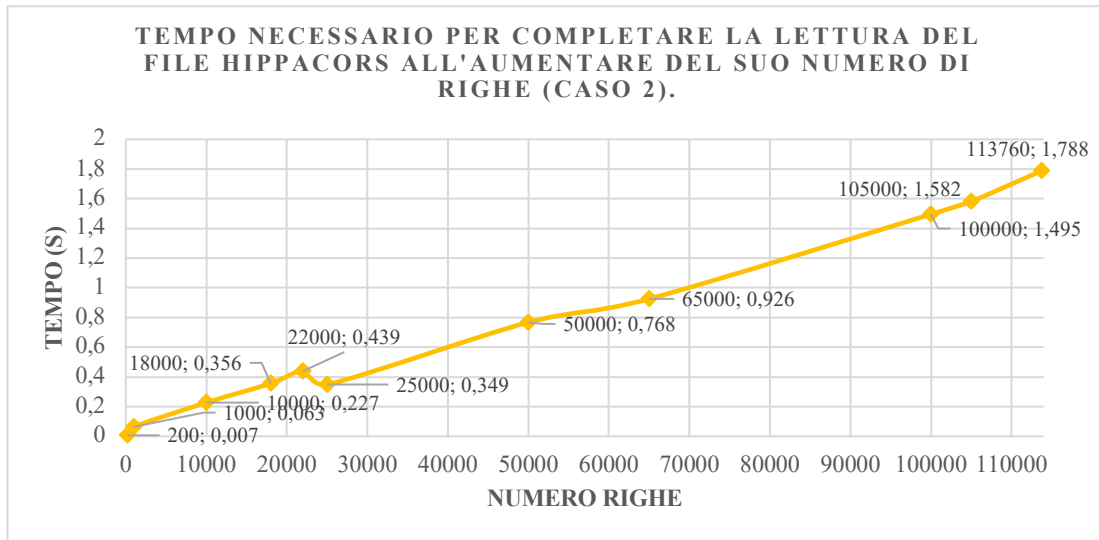


Figura 5.2 - Caricamento e lettura del file del catalogo Hipparcos caso 2.

5.1.2 Gestione del numero elevato di stelle

Essendo uno degli obiettivi dell'applicazione la rappresentazione di tutte le stelle contenute nel catalogo Hipparcos, è stato necessario trovare una soluzione opportuna che permettesse la gestione del grosso numero di oggetti nel migliore dei modi. Come discusso nel capitolo 4, sezione 4.1.1.3 si è fatto ricorso al sistema particellare, ma l'idea iniziale era differente.

Si era pensato infatti di generare tutti gli elementi della scena come `gameObject` e fare in modo che questi istanziassero il prefab opportuno sulla base della loro distanza dall'origine. Questa soluzione era basata sul concetto di LOD (Level Of Detail), tecnica che crea e renderizza un oggetto con diverso livello di dettaglio a seconda della distanza tra l'oggetto stesso e la camera, tra l'oggetto e l'origine nel caso di Astra Data Navigator.

Tuttavia se questa tecnica era stata pensata con lo scopo di ridurre il calcolo computazionale, in realtà si è rivelata essere svantaggiosa. Essendo la distanza tra oggetto e origine calcolata ad ogni frame ed essendo gli oggetti che richiedevano tale gestione di un numero troppo elevato, il metodo del LOD andava a sovraccaricare in maniera notevole la CPU e rendeva il caricamento della scena e l'utilizzo dell'applicazione fin troppo lenti. Ad ogni frame era necessario modificare il prefab: caricando la sfera base di Unity per le stelle molto vicine all'origine e un billboard rappresentato tramite sprite per le stelle lontane.

Sono state fatte svariate prove differenti tra di loro per il numero di `gameObject` da gestire in questo modo e si è notato che questa tecnica riusciva a gestire in maniera abbastanza veloce e leggera un numero massimo di `gameObject` pari a 1000.

5.2 Test di valutazione soggettivi

Al termine della fase di sviluppo il lavoro è stato sottoposto ad un gruppo di 10 persone composto da sei uomini e quattro donne al fine di ottenere una valutazione soggettiva e dei suggerimenti per eventuali sviluppi futuri. Il gruppo è costituito da persone di età compresa tra i 25 e i 38 anni e tra di loro vi erano portatori di occhiali; caratteristica utile per avere una valutazione più precisa riguardo i fastidi che la stereoscopia 3D può generare. I partecipanti sono stati scelti tra persone con competenze diverse ma tutti hanno già vissuto una qualche esperienza nell'ambito della realtà virtuale.

Durante la prova è stato chiesto agli utenti di valutare le funzionalità dell'applicazione. A ciascun utente sono stati assegnati due compiti da svolgere: l'esplorazione dell'universo tramite GUI dunque tramite selezione delle stelle e tramite ricerca per corpi del sistema solare, e l'esplorazione del sistema solare tramite tastiera. Al termine delle prove è stato infine chiesto agli utenti di rispondere ad un questionario (Appendice A.6) in modo tale da poter stilare un'analisi statistica dei risultati ottenuti.

Il questionario si compone di tre parti. La prima parte, utile per identificare il tipo di utente, è composta da domande di carattere generale come età, sesso ed eventuali esperienze di realtà virtuale vissute.

La seconda parte è a sua volta suddivisa in due sottoparti ognuna contenente le stesse domande ma riferite ciascuna ad un task che l'utente aveva il compito di svolgere durante la sua performance. Le domande poste in questa sezione sono state in parte prese dal tool NASA-TLX e riguardano lo sforzo fisico e mentale a cui l'utente è stato sottoposto, il suo grado di soddisfazione provato al completamento di ogni task, il tempo servitogli per portare a termine i compiti, ed il senso di nausea eventualmente provato [17]. Del tool non è stata presa in considerazione la domanda relativa allo sforzo: "Quanto hai dovuto lavorare (mentalmente e fisicamente) per raggiungere il tuo livello di performance?" in quanto, in relazione al tipo di applicazione e ai fattori che si volevano valutare, questa sarebbe potuta risultare come una ripetizione delle domande relative allo sforzo fisico e mentale.

Dati gli elementi da analizzare (sforzo mentale, sforzo fisico, necessità di tempo, soddisfazione della performance, fastidio provato) questi sono stati associati a due a due, tenendo in considerazione tutte le possibili combinazioni, in modo tale da ottenere 10 coppie. Prima di poter valutare la propria performance è stato chiesto all'utente di selezionare da ciascuna delle 10 coppie il fattore che ha contribuito maggiormente al carico di lavoro, e solo dopo aver scelto i pesi dei fattori, l'utente ha potuto valutare il suo punteggio su una scala di intervallo che va da 0 a 19. In questo modo è stato così possibile analizzare due potenziali fonti di variabilità inter-rater: le differenze nella definizione del carico di lavoro tra i partecipanti e la differenza del carico di lavoro tra i compiti. Al fine di determinare il carico di lavoro mentale (MWL) soggettivo si sono moltiplicati i punteggi dell'utente per il peso relativo a quel fattore, si sono sommati tutti i valori ottenuti e divisi per la somma totale dei pesi.

La figura 5.3 mostra i risultati riguardo la valutazione soggettiva del carico di lavoro mentale (MWL) relativa ad ogni singolo task.

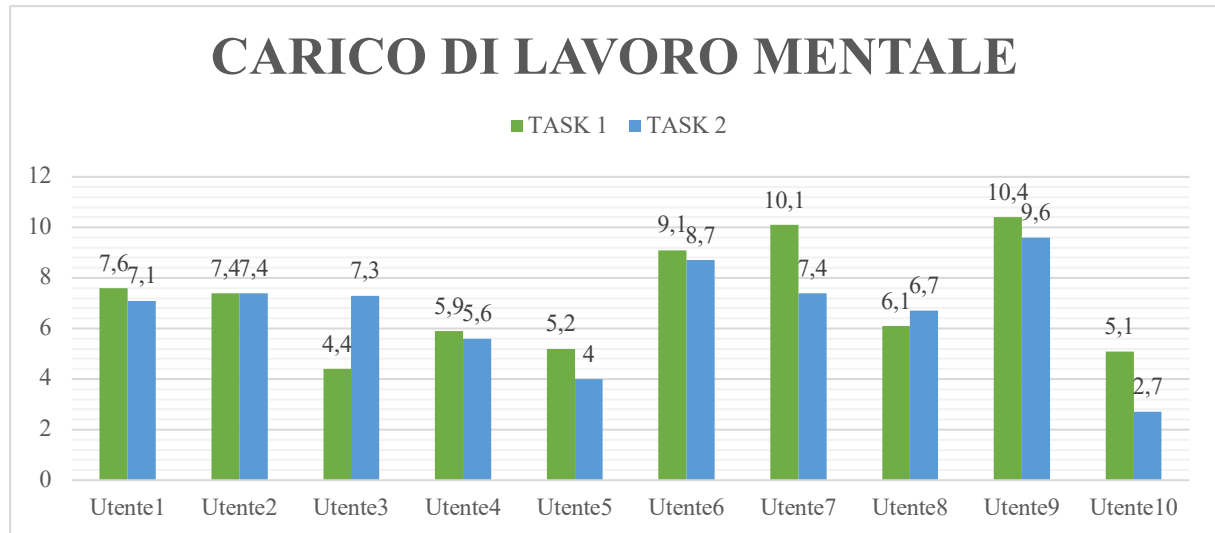


Figura 5.3 – Valutazione soggettiva del carico di lavoro mentale.

Infine la terza parte aveva lo scopo di comprendere l'opinione degli utenti riguardo intuitività, usabilità e realismo del sistema, e raccogliere opinioni riguardo il lavoro complessivo dando la possibilità all'utente di esprimere eventuali suoi suggerimenti.

Anche durante la terza parte del test l'utente ha espresso il suo giudizio tramite una scala di valutazione che va da 0 a 19 suddivisibile in quattro intervalli, ma questa volta con significato diverso.

- 0 – 4 : Scadente
- 5 – 9: Abbastanza buono
- 10 – 14: Buono
- 15 – 19: Ottimo

Come primo aspetto si è valutata l'intuitività del sistema e, dal momento che nessuno dei partecipanti aveva mai utilizzato questa applicazione, il risultato può ritenersi positivo. Il 50% degli utenti ha dato una valutazione compresa tra 5 e 9 ritenendo che il sistema sia abbastanza intuitivo mentre il restante 50% ha dato una valutazione compresa tra 10 e 14 (figura 5.4).

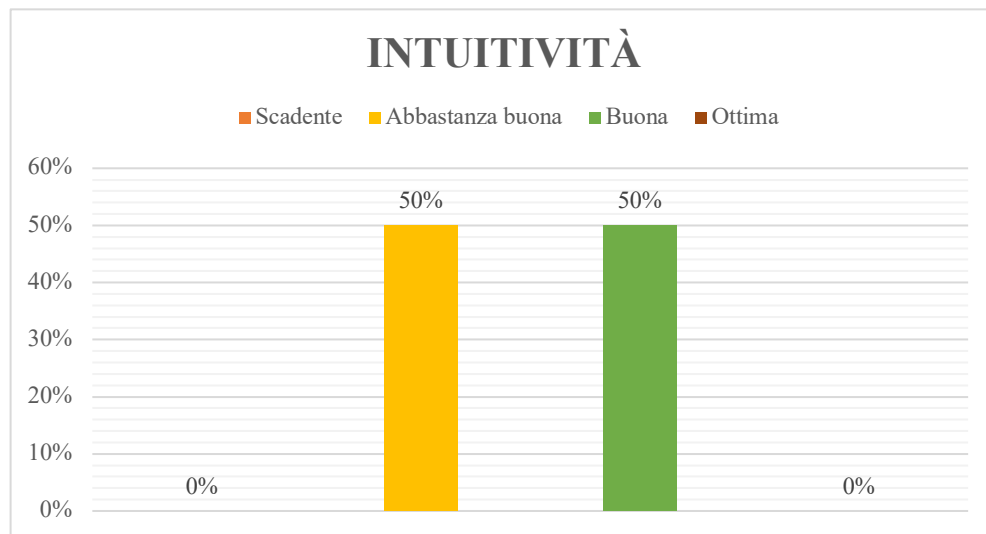


Figura 5.4 – Grafico per la valutazione dell'intuitività.

Di fondamentale importanza per qualsiasi prodotto software è che questo venga valutato usabile dagli utenti. Per usabilità si intende la facilità con cui l'utente riesce a svolgere un compito assegnatogli coerente con le funzioni offerte dal software stesso. Dai dati raccolti si osserva che tra i partecipanti il 40% ha ritenuto che il sistema abbia un'usabilità abbastanza buona, il 50% ha ritenuto che il sistema abbia una buona usabilità e il 10% ha giudicato invece l'usabilità ottima (figura 5.5).

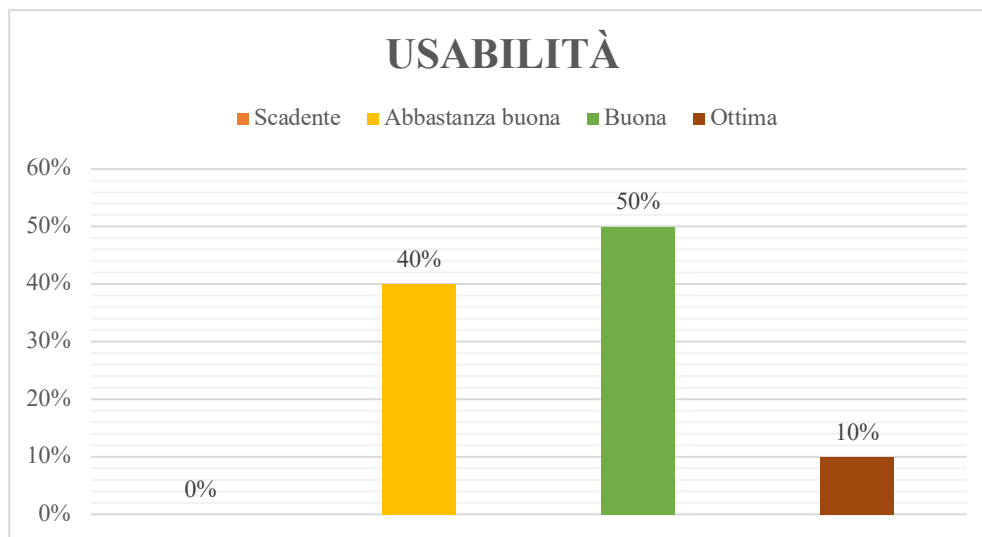


Figura 5.5 – Grafico per la valutazione dell'usabilità.

Successivamente, dato che uno degli obiettivi principale del progetto era quello di offrire all'utente una rappresentazione dei corpi celesti che risultasse il più realistica possibile, è stato analizzato anche questo aspetto. Nonostante al termine della loro esperienza alcuni utenti hanno dichiarato di gradire una maggior precisione nelle dimensioni dei satelliti rispetto ai pianeti, i risultati ottenuti permettono di affermare che questo obiettivo può dirsi raggiunto (figura 5.6).

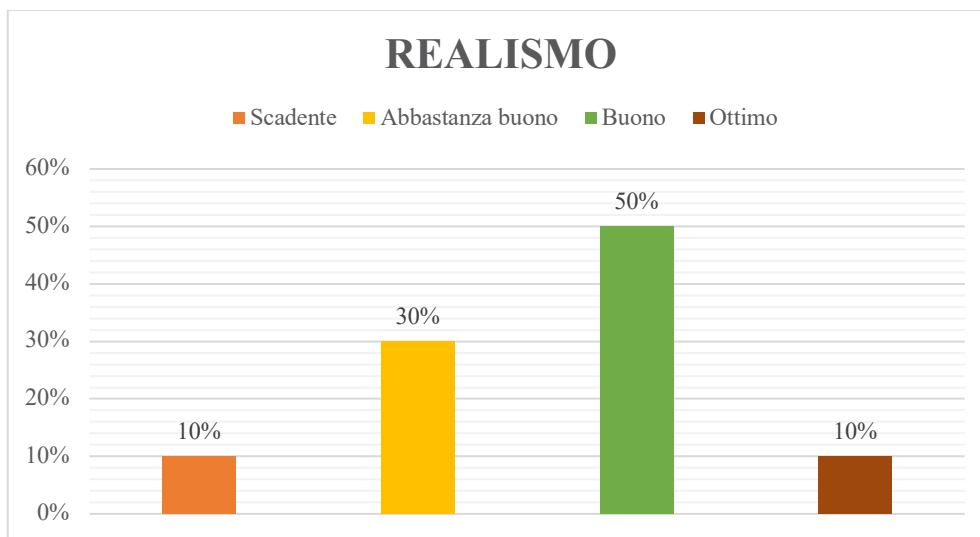


Figura 5.6 – Grafico per la valutazione del realismo.

Quando si realizzano applicazioni di realtà virtuale è necessario prestare attenzione agli eventuali problemi quali fastidio agli occhi, mal di testa e nausea che possono presentarsi all'utente a seguito di un uso prolungato della tecnologia basata su stereoscopia 3D. Come nei casi precedenti l'utente ha espresso una propria valutazione attraverso un voto compreso tra 0 e 19 e traducibile secondo i seguenti criteri:

- 0 – 4 : Nullo
- 5 – 9: Minimo
- 10 – 14: Abbastanza
- 15 – 19: Tanto

I risultati raggiunti possono ritenersi più che buoni, il 70% degli utenti non ha provato alcun senso di nausea e solo una piccola percentuale ha dichiarato di aver avuto un leggero affaticamento agli occhi (figura 5.7).

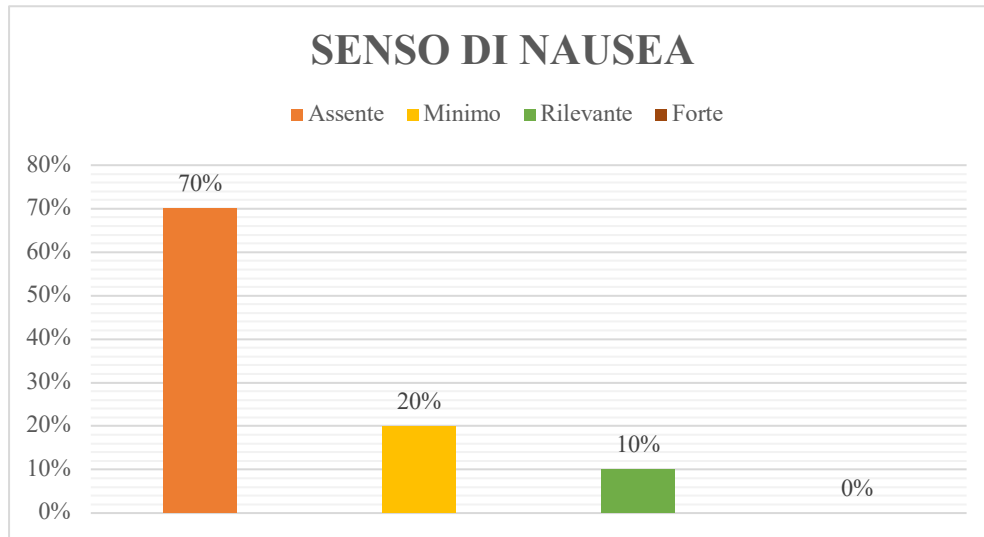


Figura 5.7 – Grafico per la valutazione del senso di nausea.

Infine, l'ultima parte del questionario chiedeva agli utenti di esprimere eventuali loro suggerimenti per miglioramenti e sviluppi futuri. Come mostra il grafico in figura 5.8 più della metà dei partecipanti si è dimostrato propenso al miglioramento degli aspetti legati alla navigazione ad esempio tramite l'introduzione di marker e sistemi di riferimento che aiutino l'utente ad orientarsi all'interno della scena.

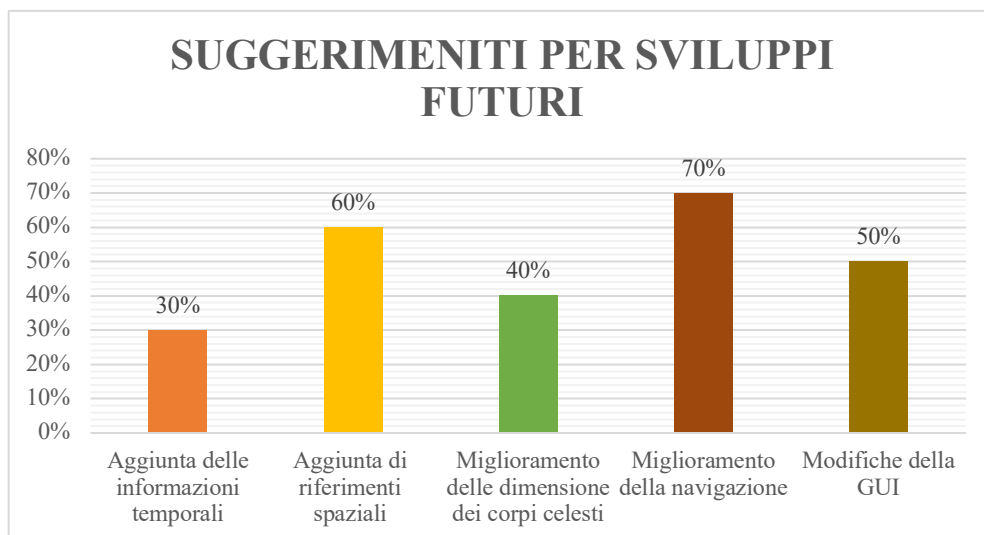


Figura 5.8 – Grafico dei suggerimenti futuri per in miglioramento di Astra Data Navigator.

CAPITOLO 6

Conclusioni e Sviluppi futuri

Il lavoro di tesi ha permesso lo sviluppo di un'applicazione di realtà virtuale che rende possibile la navigazione e l'esplorazione realistica dell'Universo. Grazie alla scalabilità offerta dall'architettura e alla facilità d'uso dell'editor Unity è possibile espandere il sistema attraverso l'inserimento di nuovi corpi celesti e cataloghi stellari. La realizzazione della scena tramite la tecnica del floating origin e i meccanismi offerti dall'asset Space Graphics Toolkit hanno reso possibile l'utilizzo di dati reali come distanze, dimensioni, velocità di rotazione etc. rendendo l'applicazione un buon strumento per analizzare e confrontare i dati cartacei con ciò che è possibile osservare sul campo al fine di rilevare nuove proprietà.

Analizzando i suggerimenti proposti dagli utenti al termine della fase di test dell'applicazione, diversi possono essere gli spunti da tenere presente per eventuali sviluppi futuri.

Una delle proposte di modifica maggiormente meritevole di essere tenuta in considerazione consiste nell'introduzione di sistemi di riferimento e nella possibilità di visualizzare le coordinate spaziali in cui si trova la camera in modo tale da aiutare l'utente ad orientarsi. L'inserimento di marker o frecce che indichino la posizione dei principali corpi celesti rispetto alla posizione dell'utente potrebbero essere una soluzione ottimale per aiutare l'utente a comprendere maggiormente la sua posizione all'interno dell'universo e quindi per aumentare il suo senso di immersione migliorando la sua esperienza di esplorazione.

La possibilità di inserire data e ora e l'utilizzo di space kernel, che permette di ottenere le posizioni reali di pianeti e satelliti in relazione ad un riferimento temporale, il miglioramento dei moti di rivoluzione grazie alle equazioni di Eulero, il perfezionamento della scala adottata per rappresentare i satelliti e l'introduzione di asteroidi, galassie, nebulose e ulteriori cataloghi stellari, ad esempio Gaia, renderebbero l'applicazione più realistica e completa e dunque più adatta ad essere impiegata come strumento di studio.

Durante i test, uno dei problemi riscontrati riguarda il metodo di selezione e avvicinamento al target selezionato. In alcuni casi la camera non riesce a posizionarsi perfettamente in posizione frontale rispetto l'oggetto e di conseguenza l'utente deve procedere con lo spostamento del suo punto di vista, quindi della camera, tramite mouse e tastiera. Un metodo che garantisca la visualizzazione dell'oggetto scelto in posizione centrata rispetto alla camera rappresenta sicuramente una delle modifiche da effettuare in quanto potrebbe permettere di ridurre anche quel minimo senso di fastidio agli occhi che alcuni utenti, seppur in minima percentuale, hanno dichiarato di percepire durante la simulazione. Sempre in relazione a questo tipo di fastidio si potrebbe inoltre rendere più fluido il movimento della camera soffermandosi con particolare attenzione sul moto di rivoluzione dei pianeti, un meccanismo che renda più fluido e meno scattoso il movimento sarebbe preferibile.

Altri possibili sviluppi futuri che potrebbero portare al miglioramento e alla possibilità di offrire all'utente un'esperienza più completa sono:

- La creazione di un'interfaccia che permetta di scegliere se caricare i dati da file o da database, dunque la possibilità di leggere i dati da database in modo da incrementare le prestazioni del sistema riducendo i tempi di lettura e caricamento delle informazioni necessarie per istanziare e gestire i modelli.

- La possibilità di ottenere informazioni sulle stelle non solo riguardo le loro caratteristiche tecniche ma anche sulle loro proprietà come fase della vita in cui si trovano ed altre curiosità.
- L'opzione di cambiare la scala spaziale e le unità di misura della velocità quando ci si sposta.

Lo studio presentato mostra come la realtà virtuale possa essere di grande supporto allo studio di dati e informazioni scientifiche. Crea immagini che rendono il processo di apprendimento più veloce. È possibile dunque concludere che nonostante occorra ancora investire, studiare, sviluppare e apportare miglioramenti, questa tecnologia, sebbene forse ancora troppo costosa e inesplorata, sta dimostrando di possedere enormi potenzialità.

APPENDICE

A.1 FloatingScaleParticle: ResizeSystem

```
public double ResizeSystem(Vector3 systemPosition)
{
    Vector3 distanceSys = (new Vector3((float)(SgtFloatingOrigin.CurrentPoint.P
osition.LocalX), (float)(SgtFloatingOrigin.CurrentPoint.Position.LocalY),
(float)(SgtFloatingOrigin.CurrentPoint.Position.LocalZ)) - systemPosition);
    double distance = System.Math.Sqrt(distanceSys.x * distanceSys.x + distance
Sys.y * distanceSys.y + distanceSys.z * distanceSys.z);
    if (distance <= DistanceMin)
    {
        linearS = 1500;
    }
    else
    {
        var distanceRange = DistanceMax - DistanceMin;

        distance -= (float)DistanceMin;
        if (distance >= distanceRange)
        {
            distance = (float)distanceRange * 0.5f;
        }
        else
        {
            var distance01 = distance / distanceRange;
            distance -= (float)(distance * 0.5 * distance01);
        }

        linearS = (distance * ScaleMultiplier);
    }
    return linearS;
}
```

A.2 ParticleGenerator: Add

```
public void Add(SgtPosition position, Color32 color, float size)
{
    ParticleSystem.Particle particle = new ParticleSystem.Particle();
    particle.position -= (new Vector3((float)((-
position.LocalX / CellSize) * CellSize), (float)((-
position.LocalY / CellSize) * CellSize),
(float)((-position.LocalZ / CellSize) * CellSize)));

    particle.startColor = color;

    scaleP = (float)FloatingScaleParticle.Instance.ResizeParicle(particle.posit
ion);
    particle.startSize = (float)(size * scaleP);

    _particles.Add(particle);
}
```

A.3 FloatingScaleParticle: ResizeParticle

```

public double ResizeParicle(Vector3 particlePosition)
{
    Vector3 distancePart = (new Vector3((float)SgtFloatingOrigin.CurrentPoint.P
osition.LocalX, (float)SgtFloatingOrigin.CurrentPoint.Position.LocalY,
(float)SgtFloatingOrigin.CurrentPoint.Position.LocalZ) - particlePosition);
    double distance = System.Math.Sqrt(distancePart.x * distancePart.x + distan
cePart.y * distancePart.y + distancePart.z * distancePart.z);
    if (distance <= DistanceMin)
    {
        linearP = 0;
    }
    else
    {
        var distanceRange = DistanceMax - DistanceMin;
        distance -= (float)DistanceMin;
        if (distance >= distanceRange)
        {
            distance = (float)distanceRange * 0.5f;
        }
        else
        {
            var distance01 = distance / distanceRange;
            distance -= (float)(distance * 0.5 * distance01);
        }
        linearP = (distance * ScaleMultiplier);
    }
    return linearP;
}

```

A.4 StarInfoPanel

```

private void Start()
{
    Disable();
    rectT = GetComponent<RectTransform>();

    EventTrigger trigger = GetComponent<EventTrigger>();
    EventTrigger.Entry entry = new EventTrigger.Entry();
    entry.eventID = EventTriggerType.Drag;
    entry.callback.AddListener((data) => { OnDragDelegate((PointerEventData)dat
a); });
    trigger.triggers.Add(entry);
}

public void OnDragDelegate(PointerEventData data)
{
    Camera cam = this.GetComponentInParent<Canvas>().worldCamera;
    Ray ray = cam.ScreenPointToRay(data.position);
    Vector3 rayPoint = ray.GetPoint(Vector3.Distance(transform.position, cam.tra
nsform.position));
    transform.position = new Vector3(rayPoint.x, rayPoint.y, this.GetComponentI
nParent<Canvas>().planeDistance);
}
}

```

A.5 FloatingScalerPlanet: SetDistance

```
private void SetDistance(double distance)
{
    var distance0 = SgtPosition.Distance(this.GetComponent<SgtFloatingPoint>().
Position, Camera.main.GetComponent<SgtFloatingPoint>().Position);
    if (ScaleByDistance == true)
    {
        if (distance0 <= DistanceMin)
        {
            transform.localScale = Vector3.zero;
        }
        else
        {
            var distanceRange = DistanceMax - DistanceMin;
            distance0 -= DistanceMin;
            if (distance0 >= distanceRange)
            {
                distance0 = distanceRange * 0.2f;
            }
            else
            {
                var distance01 = distance0 / distanceRange;
                distance0 -= distance * 0.2 * distance01;
            }
            var linear = distance0 * ScaleMultiplier;
            transform.localScale = Scale * (float)linear;
        }
    }
    else
    {
        transform.localScale = Scale * (float)ScaleMultiplier;
    }
}
```

A.6 Questionario per la valutazione di Astra Data Navigator

1. Et .....

2. Sesso

Maschio ☐ Femmina ☐

3. Hai mai vissuto un'esperienza di realt  virtuale?

Si ☐

No ☐

Se s , quale?

.....

4. Esegui i task e rispondi alle domande.

Di ogni coppia seleziona il membro che rappresenta il fattore pi  influente sulla variazione del carico di lavoro in questi task.

Necessit  di tempo / Fastidio provato

Sforzo fisico / Fastidio provato

Soddisfazione della performance / Fastidio provato

Necessit  di tempo / Sforzo fisico

Sforzo fisico / Soddisfazione

Necessit  di tempo / Sforzo mentale

Soddisfazione / Sforzo mentale

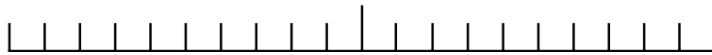
Soddisfazione / Necessità di tempo

Sforzo mentale / Sforzo fisico

Fastidio provato / Sforzo mentale

Task 1: Partendo dal sole, attraverso la GUI, esplora l'universo spostandoti tra stelle, pianeti e satelliti.

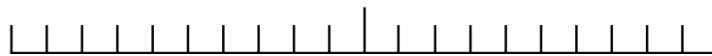
1.1 Quanto sforzo mentale hai utilizzato?



Veramente poca

Veramente tanta

1.2 Quanto sforzo fisico hai dovuto impiegare?



Veramente poca

Veramente tanta

1.3 Quanto è stato veloce il compito?



Molto basso

Molto alto

1.4 Quanto sei soddisfatto della tua performance?



Veramente poco

Veramente tanto

1.5 Quanto fastidio, senso di nausea hai provato?



Veramente poco

Veramente tanto

Task 2: Posizionati sulla terra ed esplora i pianeti e i satelliti vicini grazie all'utilizzo della tastiera.

2.1 Quanto sforzo mentale hai utilizzato?



Veramente poca

Veramente tanta

2.2 Quanto sforzo fisico hai dovuto impiegare?



Veramente poca

Veramente tanta

2.3 Quanto è stato veloce il compito?



Molto basso

Molto alto

2.4 Quanto sei soddisfatto della tua performance?



Veramente poco

Veramente tanto

2.5 Quanto fastidio, senso di nausea hai provato?



Veramente poco

Veramente tanto

5.Quanto sono intuitivi il funzionamento e l'interfaccia?



Veramente poco

Veramente tanto

6.Quanto reputi realistica la rappresentazione degli oggetti astronomici?



Veramente poco

Veramente tanto

7.Quanto questo tipo di applicativo può essere utile nel tuo ambito lavorativo?



Veramente poco

Veramente tanto

8.Valutazione usabilità del sistema.



Scadente

Ottima

9. Suggerimenti

.....
.....

Bibliografia

- [1] URL: <https://arvrjourney.com/virtual-reality-2019-vr-industry-enters-high-growth-adoption-phase-82843d8cc41a>
- [2] URL: https://www.accenture.com/us-en/insights/technology/responsible-immersive-technologies?utm_source=ActiveCampaign&utm_medium=email&utm_content=Innovation+Focus+%234+La+realt%C3%A0+virtuale+non+%C3%A8+solo+console+%F0%9F%8E%AE%2C+tutti+i+casi+concreti+e+gli+esempi+in+azienda&utm_campaign=NSL+EICS+%234
- [3] URL: <https://www.realmore.net/cose-la-realta-virtuale-come-funziona-ed-ambiti-di-applicazione/>
- [4] Karl Hille, Gen. 2020, Better than Reality: NASA Scientists Tap Virtual Reality to Make a Scientific Discovery. URL: <https://www.nasa.gov/feature/goddard/2020/scientists-tap-virtual-reality-for-discovery>
- [5] Lynn Jenner, Aug. 2017, NASA Explores Potential of Altered Realities for Space Engineering and Science. URL: <https://www.nasa.gov/feature/goddard/2017/nasa-explores-potential-of-altered-realities-for-space-engineering-and-science>
- [6] URL: <https://chandra.harvard.edu/vr/>
- [7] Chen, C. H., Yang, J. C., Shen, S., & Jeng, M. C., Lug. 2007, A Desktop Virtual Reality Earth Motion System in Astronomy Education
- [8] URL: <https://en.wikipedia.org/wiki/SpaceEngine>
- [9] URL: <https://en.wikipedia.org/wiki/Celestia>
- [10] Burdea G., Coifet P., 2003, Virtual Reality Technology (2nd ed.)
- [11] J.P. Hale, Ott. 1996, Applied virtual reality in aerospace. DOI: 10.1109/WESCON.1996.554573
- [12] URL: <https://techcrunch.com/2020/01/29/nasa-finds-real-uses-for-vr-and-ar-in-astronomy-and-engineering/>
- [13] URL: https://it.wikipedia.org/wiki/IEEE_754
- [14] C. Thorne, 2007, Origin-centric techniques for optimising scalability and the fidelity of motion, interaction and rendering
- [15] URL: https://www.dropbox.com/s/4a7nbqi2w3ocnr2/Vampirasu%20Studios.zip?dl=0&file_subpath=%2FVampirasu+Studios%2FScaleSpace+with+Floating+Origin%2FScaleSpaceManager_Manual.pdf
- [16] URL: <https://assetstore.unity.com/packages/tools/level-design/space-graphics-toolkit-4160>
- [17] URL: <https://humansystems.arc.nasa.gov/groups/TLX/>

RINGRAZIAMENTI

A conclusione di questo percorso desidero ringraziare tutte le persone che mi hanno supportata e mi hanno permesso di portare a termine questo lavoro.

Vorrei innanzitutto ringraziare il Prof. Andrea Sanna per la disponibilità riservatami in ogni momento, per i consigli e le osservazioni che mi hanno aiutata durante la stesura dell'elaborato.

Un ringraziamento va a Eugenio Topa e Alberto Alessio, sempre pronti a darmi le giuste indicazioni in ogni fase dello sviluppo del progetto, per i loro consigli e per i suggerimenti.

Ringrazio anche tutte le persone conosciute in Altec che mi hanno accompagnata durante questo percorso.

Grazie ai colleghi che hanno condiviso con me gioie, fatiche e progetti. Desidero soprattutto ringraziare Alice per avermi sempre incoraggiata, per essere sempre stata vicina e per aver condiviso con me tutti i momenti più importanti.

Un grazie speciale va alla famiglia del B-Evolution che mi ha accolta facendomi sentire una di loro.

Un grazie speciale va a Cristiano per essermi stato vicino negli ultimi due anni, per avermi insegnato a credere di più in me stessa, per avermi supportata e sopportata, e soprattutto perché continua ad esserci.

Il grazie più grande va alla mia famiglia che spero di aver reso orgogliosa. Grazie ai miei genitori che con i loro sacrifici hanno reso possibile tutto questo e perché nonostante la distanza sono sempre stati presenti. Grazie per avermi sempre spronata a dare il meglio di me, per avermi incoraggiata quando pensavo di non farcela e per essere stati comprensivi nei miei momenti di sfogo. Grazie a mio fratello che c'è sempre stato.

Infine vorrei dedicare questo traguardo a me stessa, ai miei sacrifici e alla mia tenacia che mi hanno permesso di arrivare fin qui.

Grazie.