POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria Informatica

Extended Abstract

Sviluppo di una soluzione per la rappresentazione interattiva di cataloghi stellari in Realtà Virtuale

Relatore

Prof. Andrea SANNA

Candidato

Paolo GALLO

Aprile, 2021

Indice dei Contenuti

Introduzione

Negli ultimi anni, la quantità di dati raccolti in ambito scientifico ed in particolare quello astronomico è cresciuta a dismisura, mettendo a disposizione dei ricercatori enormi moli di informazioni da dover analizzare e studiare. Questa situazione ha portato alla luce le limitazioni degli strumenti "tradizionali" di analisi, insieme alla necessità di sviluppare nuovi e moderni sistemi di visualizzazione di questi dati. Le tecniche di *Data Visualization* in ambito scientifico mirano quindi a realizzare un'esplorazione visuale ed interattiva delle informazioni, rappresentate in forma grafica, che permetta ai ricercatori di individuare tendenze, pattern ed anomalie nei dati.

In questo settore, la Realtà Virtuale (VR) ha dimostrato di essere una tecnologia in grado di fornire potenti strumenti per la creazione ambienti virtuali nel quale rappresentare graficamente un insieme di dati con un elevato grado di interattività, dando ai ricercatori la possibilità di osservare i fenomeni da prospettive nuove e aprendo le porte a scenari altrimenti impossibili, come la collaborazione a distanza all'interno di ambienti virtuali condivisi.

In aggiunta a tutto ciò, la Realtà Virtuale è anche estremamente adatta per l'utilizzo in ambito educativo, in cui ha già dimostrato di produrre risultati di apprendimento più rapidi e duraturi nel tempo, grazie al maggiore coinvolgimento ed all'elevato grado di immersività di queste esperienze.

Nell'ambito astronomico, la visualizzazione di cataloghi stellari di grandi dimensioni è attualmente un tema di elevato interesse. La missione Gaia dell'Agenzia Spaziale Europea (ESA) [1] sta raccogliendo enormi quantità di dati astrometrici ad elevata precisione ed il relativo catalogo - non ancora definitivo - conta già oltre 1.8 miliardi di oggetti distinti.

Stato dell'Arte

Rappresentare un dataset così vasto in tempo reale è una sfida capace di sovraccaricare anche l'hardware più prestazionale. Le applicazioni che, attualmente, si occupano della visualizzazione di questo tipo di dati si possono suddividere in 3 categorie in base a come approcciano questo problema:

- **Nuvole di punti:** le *point clouds* sono un insieme di punti nello spazio, ciascuno dei quali è caratterizzato da una terna di coordinate X/Y/Z e, in alcuni casi, una quarta informazione riguardante il colore.

È una tecnica molto utilizzata per la visualizzazione di grandi dataset, che trova nella compattezza e nell'efficienza i suoi punti di forza. Le nuvole di punti sono già state impiegate per rappresentare milioni di stelle in un ambiente 3D (ad esempio *Gaia 3D Starmap* [2] oppure il servizio web *Gaia Archive Visualization Service* [3]).

La principale limitazione di queste rappresentazioni consiste nello scarso livello di interazione con la scena, che spesso mostra unicamente la distribuzione statistica dei dati ed impedisce l'accesso alle informazioni riguardanti il singolo oggetto, permettendo all'utente solo di ingrandire o ruotare la nuvola di punti per osservarla meglio.

- **Planetari:** questi software sono progettati per l'utilizzo negli omonimi edifici, dotati di una grande cupola sulla quale vengono proiettate immagini della volta celeste, in cui gli oggetti si muovono in modo realistico per come appaiono ad un osservatore sulla Terra.

  Sistemi del genere (ad esempio *Stellarium* [4]) sono in grado di gestire dati riguardanti milioni di corpi celesti e forniscono un buon grado di interattività: ad esempio, permettono di ingrandire una porzione specifica del cielo per poterla analizzare meglio e di selezionare i singoli corpi celesti per ottenere maggiori informazioni a riguardo.

  Il limite principale di questi strumenti è l'ancoraggio del punto di osservazione alla superficie terrestre, che impatta negativamente la dimensionalità dei dati rappresentati.

- **Esplorazione dell'universo in VR:** questo tipo di applicazioni (tra cui *Celestia* [5], *Space Engine* [6] e *CosmoScout VR* [7]) si concentrano principalmente sull'aspetto esplorativo dell'ambiente 3D, all'interno del quale l'utente può muoversi liberamente per osservare ed interagire con gli oggetti rappresentati da una visuale in prima persona.

  Questo approccio è ideale per coniugare finalità scientifiche ed educative, in quanto permette sia ad utenti non esperti che a professionisti del settore di estrarre informazioni utili dai dati rappresentati.

  La limitazione principale di queste applicazioni risiede però nella dimensione dei cataloghi supportati. Spesso, infatti, la quantità di dati gestiti da questi software è decisamente ridotta rispetto alle controparti basate su nuvole di punti, a causa dell'elevato costo computazionale necessario per questo tipo di rappresentazione.

Obiettivi e requisiti del progetto

Nel contesto del progetto europeo NEANIAS [8], ALTEC - Aerospace Logistics Technology Engineering Company - mira allo sviluppo di un sistema per la rappresentazione interattiva di dati astronomici in Realtà Virtuale, in particolare con lo scopo di visualizzare il catalogo stellare Gaia a fini scientifici ed educativi.

L'obiettivo del lavoro di tesi è quindi quello di estendere l'applicazione *Astra Data Navigator*, sviluppata utilizzando il motore grafico Unity, introducendo un sistema in grado di caricare e visualizzare in tempo reale cataloghi stellari di grandi dimensioni (contenenti milioni o perfino miliardi di oggetti) all'interno di un ambiente 3D completamente interattivo.

Rispetto alle tecniche precedentemente descritte, questa nuova applicazione intende coniugare le moli di dati tipicamente rappresentate sotto forma di nuvole di punti con i vantaggi di un universo virtuale liberamente esplorabile in VR, in cui ogni oggetto è interagibile e può essere visualizzato in 3D.

I requisiti del progetto sono quindi i seguenti:

- **Scalabilità** a dataset contenenti decine o centinaia di milioni di oggetti, per poter supportare i più recenti e corposi cataloghi stellari.
- **Configurabilità e possibilità di estensione:** oltre al catalogo predefinito, l'applicazione permetterà la visualizzazione di qualunque insieme di dati compatibile fornito dall'utente, al quale è anche offerta la possibilità di controllare il funzionamento dello strumento tramite un sistema di configurazione.
- **Navigazione fluida ed intuitiva:** l'applicazione deve fornire all'utente degli strumenti efficaci ed intuitivi per l'esplorazione dell'ambiente virtuale e dei dati contenuti al suo interno.
- **Interazione con ogni oggetto** presente all'interno della scena, in modo che l'utente possa utilizzare gli strumenti a propria disposizione per esplorare il catalogo ed ottenere informazioni aggiuntive riguardo ad ogni corpo celeste (tra cui parametri fisici e astrometrici).
- **Realismo grafico** dell'ambiente 3D e degli oggetti rappresentati, per fornire una visualizzazione scientificamente accurata dei dati ed aumentare il senso di immersività dell'applicazione, senza sacrificarne le performance.
- **Supporto alla stereoscopia** per fornire un'esperienza più coinvolgente grazie all'utilizzo delle apparecchiature di cui è dotato il VR-LAB di ALTEC (proiettore stereoscopico ed occhiali LCD attivi).

Capitolo 1

Architettura e Sviluppo

1.1 Cataloghi stellari

La scelta di formato per rappresentare i cataloghi stellari utilizzati dall'applicazione è ricaduta sui database relazionali. Rispetto ai file di testo, formato utilizzato originariamente, questi sistemi sono molto più adatti per memorizzare ed accedere a grandi quantità di informazioni, dell'ordine di milioni di oggetti. Questi vantaggi derivano principalmente dalla codifica dei dati numerici, molto più efficiente del formato testuale, e dalla flessibilità nell'accesso alle informazioni tramite query in linguaggio SQL.

La selezione dei parametri da includere nel catalogo è stata fatta ponendo particolare attenzione all'utilizzo di memoria ed allo spazio di archiviazione necessario. Gli attributi obbligatori di un catalogo sono 6 (codice di identificazione univoco, posizione X/Y/Z, temperatura superficiale e raggio della stella), mentre i campi opzionali includono informazioni comuni riguardo alle stelle più conosciute (il nome proprio e la costellazione di appartenenza) ed ulteriori codici identificativi.

L'applicazione integra un database *SQLite* contenente un insieme di circa 2.5 milioni di stelle prelevate dal catalogo Tycho-2. Questo dataset è stato costruito prelevando le informazioni dalla più recente *data release* di Gaia (Early Data Release 3, EDR3, Dicembre 2020 [9]). Il processo è stato automatizzato tramite alcuni script Python che, utilizzando i pacchetti `astropy` e `astroquery`, sono in grado di eseguire delle query sull'archivio pubblico di Gaia per poi scaricare i dati ed inserirli nel database di destinazione. Questi script, inoltre, si occupano di effettuare la conversione delle coordinate (da equatoriali a cartesiane) e di inserire informazioni aggiuntive quali il nome della stella e la relativa costellazione, quando disponibili.

In aggiunta a questi due cataloghi, l'utente può (tramite il sistema di configurazione) specificare una sorgente aggiuntiva di dati, da cui l'applicazione attingerà in fase di caricamento per popolare la scena. Il dataset fornito in questo modo può essere un database SQL oppure un semplice file di testo (CSV), ma deve necessariamente contenere tutti i parametri obbligatori per ciascuna stella.

1.2 Sistema di configurazione

Come da requisiti, *Astra Data Navigator* offre all'utente la possibilità di modificare alcuni aspetti dello strumento per adattarlo alle proprie necessità. Questo sistema di configurazione è realizzato tramite un file XML, caricato come prima cosa all'avvio dell'applicazione, il cui contenuto viene utilizzato da molteplici componenti per regolare il proprio comportamento sulla base delle opzioni specificate.

La scelta del formato XML è dovuta principalmente alla sua semplicità ed elevata interpretabilità, che lo rendono adatto ad essere manipolato dall'utente, ma in parte anche al supporto nativo offerto dal linguaggio C# per questa codifica.

Le opzioni disponibili all'interno di questo file permettono di personalizzare aspetti come la registrazione degli eventi applicativi, la simulazione del moto di alcuni corpi celesti e, soprattutto, offrono la scelta del metodo di caricamento dei dati stellari e la possibilità di specificare un dataset aggiuntivo (oltre a quello integrato) da visualizzare all'interno dell'universo virtuale.

1.3 Caricamento dei dati

Il componente che si occupa di effettuare il caricamento dei dati dai cataloghi prende il nome di *AdnLoader*, il quale coordina il lavoro di un insieme di script di caricamento specializzati, uno per ogni possibile fonte di dati (catalogo stellare integrato, database esterno, file CSV, cataloghi di pianeti e satelliti).

Per potersi meglio adattare alle necessità degli utenti e alle diverse quantità di dati da gestire, l'applicazione offre la scelta tra due diversi metodi di accesso ai cataloghi, effettuabile tramite il file di configurazione:

- **Caricamento statico:** è la modalità operativa più "tradizionale", che prevede di eseguire gli script di caricamento una volta sola all'avvio dell'applicazione, leggendo tutti i dati disponibili dai cataloghi (fino ad un massimo di 5 milioni di oggetti) ed inserendoli nella scena. In questo modo, il tempo di avvio e l'utilizzo di risorse sono più elevati ma, una volta completata la procedura iniziale, è possibile navigare nell'ambiente 3D in modo fluido e senza interruzioni.

- **Caricamento dinamico:** per ridurre il consumo di risorse ed i tempi di accesso ai dati, soprattutto per cataloghi di grandi dimensioni, questo metodo mantiene in memoria unicamente i corpi celesti visibili dalla posizione attuale. Ciò è realizzato tramite una query di selezione, eseguita sul database, che tiene conto del rapporto tra la dimensione dell'oggetto e la sua distanza dall'osservatore. Affinché la selezione sia sempre aggiornata, questa procedura deve essere ripetuta ogni volta che la camera si sposta per più di 10 parsec[1] all'interno della scena.

[1] 1 parsec = 3.086×10^{13} km

Questo metodo riduce notevolmente la quantità di memoria richiesta dall'applicazione e supporta cataloghi con più di 5 milioni di oggetti, ma introduce delle interruzioni durante la navigazione all'interno della scena, dovute al caricamento di nuove stelle man mano che l'osservatore si sposta.

In base alla quantità di dati forniti, la procedura di avvio dell'applicazione potrebbe risultare lenta e viene quindi eseguita in maniera asincrona, mentre all'utente viene presentata una schermata di caricamento animata in cui sono visualizzate delle informazioni di avanzamento e lo stato attuale del processo.

1.4 Gestione degli oggetti

Una volta caricate, tutte le informazioni riguardanti i corpi celesti lette dai cataloghi sono memorizzate in istanze della classe *AdnCelestialObject* e dei suoi derivati (ad esempio *AdnPlanet* per i pianeti e *AdnStar* per le stelle).

Tutti questi oggetti sono contenuti all'interno di una struttura dati, centrale all'applicazione, che prende il nome di *AdnCelestialMap*. L'efficienza di questo componente è di fondamentale importanza per garantire buone prestazioni durante l'esecuzione di alcuni algoritmi e procedure di aggiornamento della scena.

L'architettura di questa struttura dati prevede l'unione di un `Dictionary` (ovvero una hash map) con un insieme di `List` (array dinamici), una per ogni categoria di corpo celeste. Le istanze di *AdnCelestialObject* sono contenute nelle liste, mentre le informazioni del dizionario permettono di risalire alla posizione di un oggetto nella relativa lista a partire dal suo identificativo. Tramite questa combinazione è possibile ottenere una complessità computazionale $\mathcal{O}(1)$ per tutte le operazioni chiave eseguite dagli algoritmi principali dell'applicazione, massimizzandone l'efficienza.

Il compito di inizializzare e, successivamente, aggiornare la scena con i dati caricati dai cataloghi è delegato al componente *AdnUniverseManager*. Questo script si occupa di creare e posizionare all'interno dell'ambiente 3D le istanze dei *Prefab* associati agli oggetti in questione. Non è pensabile istanziare un *GameObject* per ciascuna stella del catalogo, a causa dell'eccessivo carico di lavoro che comporterebbe per il motore Unity, per cui si è scelto di creare e rendere interagibili solo le stelle maggiormente visibili dalla posizione attuale dell'osservatore ed aggiornare la scena man mano che questa posizione cambia. Analogamente al caso del caricamento dinamico, anche questo criterio di selezione si basa sul rapporto tra distanza e dimensione dell'oggetto ed è realizzato dalla funzione `IsReachable()`.

1.5 Aspetti grafici

L'oggetto tridimensionale utilizzato per rappresentare una stella è basato sul modello di una sfera, disponibile all'interno di Unity, alla quale sono stati applicati il materiale animato *StarSurface* ed alcuni componenti forniti dall'asset *Space Graphics Toolkit (SGT)* [10] per realizzare effetti visivi come la corona stellare e

le prominenze. Per meglio caratterizzare ciascun oggetto, le tonalità cromatiche utilizzate per colorare la superficie e la corona della stella sono derivate, tramite calcoli scientifici, dal valore della temperatura superficiale presente nel catalogo.

Come già è stato descritto, non è pensabile istanziare all'interno della scena un *GameObject* per ciascuna stella caricata dal catalogo. Per questa ragione, il modello 3D menzionato in precedenza viene creato dall'applicazione solo quando l'osservatore si trova sufficientemente vicino all'oggetto (meno di 1 parsec di distanza).

La soluzione impiegata, invece, per rappresentare tutte le altre stelle nella scena è basata su uno shader personalizzato, gestito dal componente *AdnStarfield*. Sfruttando la tecnica del rendering[2] procedurale e l'elevata potenza di calcolo offerta dalle moderne GPU, questo shader visualizza ciascuna stella sotto forma di billboard[3] su cui è mappata la texture di un bagliore luminoso, anch'esso colorato in base alla temperatura dell'oggetto. La fase di inizializzazione dello shader prevede che la CPU riempia un *ComputeBuffer* con le informazioni di posizione, dimensione e colore di ciascuna stella, dopodichè l'hardware grafico si occupa di eseguire i calcoli su tutti questi dati. Essendo tutte operazioni estremamente parallelizzabili, risultano particolarmente adatte ad essere eseguite dall'architettura di una GPU ed infatti questo shader è in grado di renderizzare decine di milioni di stelle in tempo reale con ottime prestazioni.

Un problema incontrato nella messa in pratica di questo approccio è stata la perdita di precisione nei calcoli effettuati con variabili `float` a singola precisione all'interno dello shader, che causava la comparsa di jitter[4] negli oggetti più lontani dall'origine. Non è possibile utilizzare il tipo `double` in questi programmi, ma è stata invece adottata una soluzione che prevede l'utilizzo di due variabili a 32-bit più alcuni calcoli aggiuntivi per emulare la doppia precisione nelle operazioni in virgola mobile, ottenendo buoni risultati.

1.6 Tecniche di ottimizzazione

Secondo i requisiti del progetto, l'applicazione deve essere in grado di gestire grosse quantità di dati (e quindi di oggetti) in tempo reale, con un buon livello di performance. Oltre alle scelte già descritte (formati di memorizzazione e strutture dati efficienti, shader per il rendering delle stelle), sono state sfruttate alcune tecniche di ottimizzazione aggiuntive:

- **Multi-threading:** consiste nel suddividere il lavoro fra diversi thread[5] in modo che i dati disponibili vengano elaborati in parallelo, sfruttando l'elevato

[2]Rendering: processo di generazione di un'immagine a partire dai dati che descrivono una scena tridimensionale.

[3]Billboard: un elemento bidimensionale sempre rivolto verso l'osservatore.

[4]Jitter: sfarfallio o variazione ad alta frequenza di alcune caratteristiche dell'immagine.

[5]Thread: sottoprocessi che vengono eseguiti concorrentemente da un sistema di elaborazione.

numero di processori logici presente nelle CPU moderne (20 sulla workstation utilizzata durante lo sviluppo) per accelerare l'esecuzione degli algoritmi. Questa soluzione è adottata per determinare quali stelle devono essere istanziate nella scena e per l'inizializzazione del componente *AdnStarfield*, tramite il costrutto `Parallel.ForEach` del linguaggio C#.

- **Object pooling:** è una tecnica che mira a minimizzare il tempo speso in operazioni di allocazione della memoria e garbage collection. Per ridurre il numero di volte che queste operazioni vengono eseguite, soprattutto nelle situazioni in cui molti oggetti vengono creati e distrutti frequentemente (come nel caso dei *GameObject* contenenti i selettori delle stelle), si introduce una struttura dati in cui vengono inseriti gli oggetti non utilizzati dopo essere stati disattivati. In questo modo, quando è necessario creare un nuovo *GameObject*, viene prima controllata la *pool* per cercare se è presente un oggetto che può essere riutilizzato senza bisogno di allocarne uno nuovo.

1.7 Interfaccia utente e navigazione

I metodi di navigazione messi a disposizione dell'utente per esplorare il catalogo stellare 3D sono i seguenti:

- **Click-and-goto:** tutti gli oggetti interagibili nella scena (pianeti, satelliti e stelle raggiungibili) possono essere selezionati e, tramite un apposito bottone, è possibile raggiungerli con un'animazione di viaggio rapido (o *warp*).
- **Movimento libero:** quando non è ancorato su un oggetto 3D, l'utente ha la possibilità di direzionare liberamente la camera con il mouse e muoversi nello spazio con i tasti direzionali oppure W/A/S/D. La velocità di movimento può essere regolata tramite la rotellina del mouse, fino a valori nettamente superiori alla velocità della luce (dell'ordine di migliaia di parsec/s).
- **Pannello di ricerca:** tramite questo elemento dell'interfaccia, l'utente può ricercare un oggetto inserendone il nome (anche parziale), dopodichè ha la possibilità di effettuare il *warp* per raggiungerlo oppure, rimanendo nella posizione attuale, direzionare la visuale verso di esso con un tasto.

Per quanto riguarda l'interfaccia utente, la scelta effettuata sin dall'inizio del progetto è stata quella di mantenere un layout il più simile possibile a quello originale, introducendo però nuove funzionalità ed alcuni miglioramenti. I nuovi elementi includono le due schermate di caricamento (una mostrata all'avvio e una utilizzata nel caso di loader dinamico), un *LogPanel* per visualizzare i messaggi di log inviati dall'applicazione ed un *InfoPanel* che mostra tutte le informazioni disponibili riguardo all'oggetto selezionato. Il *SearchPanel*, invece, è stato dotato di alcune migliorie tra cui un'icona animata utilizzata per indicare lo stato della ricerca in corso, che nel caso di cataloghi contenenti milioni di oggetti potrebbe impiegare alcuni secondi.

1.8 Stereoscopia

Il supporto alla stereoscopia in *Astra Data Navigator* è stato realizzato tramite l'integrazione del plug-in XR di Unity, configurato per la piattaforma Windows Mixed Reality. Senza necessitare di ulteriori operazioni, questo componente avvia automaticamente l'applicazione in modalità stereoscopica quando rileva l'hardware adatto ed un display compatibile.

Il resto della configurazione dell'effetto stereoscopico è stata effettuata tramite il pannello di controllo dei driver *NVIDIA*, da cui si può regolare la separazione tra le immagini dei due occhi (abbassata, in questo caso, al valore di 10% per ridurre l'affaticamento della vista degli utenti) ed è anche possibile scambiare le due immagini per invertire l'effetto di tridimensionalità della scena.

Capitolo 2

Risultati

Al termine del lavoro di sviluppo dell'applicazione sono state condotte alcune analisi per valutare i risultati ottenuti, in particolar modo confrontandoli con i requisiti iniziali del progetto.

2.1 Analisi prestazionale

Il primo importante aspetto del lavoro che è stato analizzato in questa fase consiste nel confronto tra le performance della versione originale dell'applicativo e la nuova versione dello strumento *Astra Data Navigator*. Questa analisi comparativa ha evidenziato un notevole miglioramento delle prestazioni del sistema (di un fattore compreso tra 6 e 7x), dovuto alla combinazione di tutte le tecniche descritte in precedenza, in termini di memoria utilizzata, tempi di caricamento dei cataloghi e framerate[1] dell'applicazione. Il risultato è un incremento di circa due ordini di grandezza del numero di stelle che il sistema è in grado di gestire in tempo reale.

Analizzando invece il profilo prestazionale dei due metodi di caricamento dei dati, sono state evidenziate le loro diverse caratteristiche ma soprattutto le relative limitazioni. Infatti, il loader statico è limitato ad un massimo di 5 milioni di oggetti e non può scalare oltre questo valore, per il quale però offre performance decisamente buone. La modalità di caricamento dinamica, invece, riduce notevolmente il consumo di memoria dell'applicazione ed è in grado di scalare a dataset di qualunque dimensione, ma il tempo necessario per effettuare la query di selezione degli oggetti visibili aumenta linearmente con il numero di stelle nel catalogo, diventando non più sostenibile una volta superata la soglia dei 20 milioni di stelle. Inoltre, in questa modalità, il caricamento non avviene solo all'avvio dell'applicazione ma ogni volta che l'osservatore si sposta (di una distanza considerevole) nella scena, interrompendo la navigazione dell'ambiente 3D con lunghe pause che - per cataloghi molto grandi - rendono l'esperienza decisamente poco fruibile.

[1] Framerate: frequenza di riproduzione dei fotogrammi di un video o di un'animazione digitale.

2.2 Test di valutazione soggettivi

L'applicazione è stata anche fatta provare ad un gruppo di 12 persone (scelte in modo da avere un campione statistico variegato), a cui è stato poi richiesto di compilare un questionario mirato ad ottenere un riscontro sull'usabilità del sistema ed una valutazione del lavoro realizzato.

I risultati di questi test sono stati per la maggior parte positivi, in particolare per quanto riguarda le performance ed il livello di realismo grafico dell'applicazione, che sono stati valutati molto buoni da quasi tutti gli utenti. La determinazione del carico di lavoro percepito durante lo svolgimento dei task proposti, effettuato secondo il modello NASA Task Load Index (TLX) [11], ha anch'esso dato risultati positivi, insieme alla valutazione soggettiva di intuitività ed usabilità del sistema. Sotto questo aspetto, però, gli utenti hanno fornito vari suggerimenti per il miglioramento dell'attuale interfaccia, che potranno essere messi in atto durante futuri sviluppi dell'applicazione.

La richiesta di confrontare l'esperienza offerta da ciascuno dei due metodi di caricamento disponibili ha evidenziato la bontà della formula utilizzata per la selezione delle stelle visibili, in quanto quasi nessun utente ha notato una differenza visiva tra le due soluzioni. In termini di prestazioni, però, la preferenza espressa è andata nettamente in favore del loader statico, mentre i continui caricamenti introdotti dalla modalità dinamica sono stati reputati dannosi per la navigazione dalla maggior parte degli utenti.

Capitolo 3

Conclusioni

Il progetto di tesi ha portato alla realizzazione di un sistema in grado di caricare, visualizzare e gestire in tempo reale le informazioni riguardanti milioni di stelle, prelevate da appositi cataloghi astronomici.

La scalabilità di questa soluzione, nonostante sia notevolmente migliorata rispetto alla versione iniziale, non è comunque sufficiente per riuscire a rappresentare l'intero catalogo Gaia (1.8 miliardi di oggetti) in un ambiente 3D interattivo. Le prestazioni saranno sempre un aspetto problematico nello sviluppo di questi sistemi, che devono essere in grado di gestire enormi moli di dati, ma è possibile, almeno parzialmente, aggirare queste limitazioni tramite tecniche di ottimizzazione avanzate, alcune delle quali sono state utilizzate anche in questo lavoro.

Gli sviluppi futuri di questa applicazione potranno concentrarsi sul miglioramento e l'estensione dell'interfaccia utente, grazie anche ai suggerimenti ricevuti durante i test soggettivi, ma soprattutto sull'introduzione di nuove funzionalità per l'analisi dei dati. In questo senso, sarebbe molto utile realizzare un sistema per il filtraggio in tempo reale degli oggetti visualizzati sulla base di criteri personalizzati, che permetta di selezionare un sottoinsieme di dati d'interesse ed eventualmente esportare i risultati su file per usarli in seguito.

In conclusione, questo lavoro ha mostrato le potenzialità delle tecnologie di Realtà Virtuale sia come supporto alla ricerca scientifica in ambito astronomico che come strumento educativo, grazie alla visualizzazione in tempo reale di cataloghi stellari di grandi dimensioni all'interno di un ambiente 3D interattivo. L'esperienza utente potrà essere ulteriormente perfezionata ed in futuro si potranno aggiungere nuove funzionalità allo strumento, ma i risultati ottenuti sono già da ritenersi piuttosto soddisfacenti.

Riferimenti

[1] *Gaia overview*. ESA. URL: https://www.esa.int/Science_Exploration/Space_Science/Gaia/Gaia_overview (cit. a p. 1).

[2] *Gaia 3D Starmap*. URL: https://charliehoey.com/threejs-demos/gaia_dr1.html (cit. a p. 2).

[3] *Gaia Archive Visualization Service*. URL: https://gea.esac.esa.int/archive/visualization (cit. a p. 2).

[4] *Stellarium (sito web)*. URL: https://stellarium.org/ (cit. a p. 2).

[5] *Celestia (sito web)*. URL: https://celestia.space/ (cit. a p. 2).

[6] *Space Engine (sito web)*. URL: http://spaceengine.org/ (cit. a p. 2).

[7] *CosmoScout VR (repository)*. URL: https://github.com/cosmoscout/cosmoscout-vr (cit. a p. 2).

[8] *NEANIAS (sito web)*. URL: https://www.neanias.eu/ (cit. a p. 3).

[9] Gaia Collaboration, A. G. A. Brown, A. Vallenari, T. Prusti, J.H.J. de Bruijne, et al. «Gaia Early Data Release 3. Summary of the contents and survey properties». In: (ott. 2020). DOI: 10.1051/0004-6361/202039657 (cit. a p. 4).

[10] *Space Graphics Toolkit (Unity Asset Store)*. URL: https://assetstore.unity.com/packages/tools/level-design/space-graphics-toolkit-4160 (cit. a p. 6).

[11] *NASA Task Load Index (TLX) Website*. NASA. URL: https://humansystems.arc.nasa.gov/groups/TLX/ (cit. a p. 11).