



UNIVERSITÀ DI PISA

Corso di Laurea Specialistica in Informatica Umanistica

RELAZIONE

Information Landscapes: studi di usabilità e sviluppo di un sistema di authoring web

Candidato: *Daniele Duranti*

Relatori: *Ing. E. Ruffaldi*

Dott. C. Evangelista

Ing. M. Carrozzino

Anno Accademico 2011-2012

Alla mia famiglia

Indice

Introduzione	1
1 Stato dell'arte	4
1.1 Per un inquadramento dell'argomento	5
1.1.1 Gli Information Landscapes come caso specifico di realtà virtuale	5
1.1.2 I paesaggi di informazione nell'ambito dell'Information Visualization	8
1.1.3 Graphic Design e ILs: La tipografia virtuale	11
1.2 Applicazioni esistenti	13
1.2.1 Typographic Space (MIT)	13
1.2.2 Lab. Percro (Scuola superiore S. Anna)	16
1.2.3 Xanadu Space (Ted Nelson)	22
1.2.4 Un IL per la valorizzazione e divulgazione della storia dell'arte	24
1.2.5 WWW3D: un esempio di web browser 3D	26
1.2.6 City of News	27
1.3 Tentativo di Classificazione	29
1.4 Attuali ambiti di applicazione	31
1.5 Vantaggi della rappresentazione spaziale delle informazioni	32
1.6 Impostazione del lavoro di tesi	35
2 Analisi del visualizzatore XVR di Information Landscape (Lab. PERCRO)	37
2.1 La tecnologia XVR	37
2.2 Il visualizzatore XVR di Information Landscapes	40
2.2.1 La cartella di progetto: organizzazione del codice	40
2.2.2 Il diagramma delle classi	40
2.2.3 Gestione del testo	43
2.2.4 Gestione del rendering	44
2.2.5 Tecnica per il rilevamento della selezione	44
2.2.6 Contesti di fruizione e dispositivi di interazione supportati	45
2.2.7 Caricamento di un paesaggio di informazione: il file ilxb	47
3 Sviluppo di un sistema di editing web per la costruzione di Information Landscapes	50
3.1 Vantaggi di un sistema di editing web	51
3.2 Descrizione di alto livello del software realizzato	51

3.2.1	La 3D view	52
3.2.2	Modalità di editing, modalità di viewer.....	52
3.2.3	Selezione di un elemento	53
3.2.4	La vista struttura	53
3.2.5	Controlli di navigazione	54
3.2.6	Attivatori di visualizzazione assi globali e gizmos	54
3.2.7	Pannello di editing	57
3.2.8	L'interfaccia di caricamento, salvataggio e cancellazione di un IL.....	61
3.2.9	L'Interfaccia di ricerca	61
3.3	Descrizione dell'implementazione	65
3.3.1	Tecnologie e linguaggi utilizzati	66
3.3.2	Definizione di un protocollo per lo scambio dei dati da XVR a pagina web e viceversa.....	67
3.3.3	Modalità di editing, modalità di viewer.....	70
3.3.4	Creazione di un indice degli elementi dell'IL: la vista di struttura.....	71
3.3.5	Selezione di un elemento	72
3.3.6	Creazione e cancellazione di elementi.....	73
3.3.7	Manipolazione degli elementi: traslazione, scalatura, rotazione.....	77
3.3.8	Salvataggio dell'IL su server web	81
3.3.9	Caricamento e cancellazione di un IL da server web	83
3.3.10	Sviluppo della funzionalita' di ricerca full text	84
4	Usabilità del visualizzatore XVR di paesaggi di Informazione	91
4.1	Usabilità: aspetti teorici.....	92
4.1.1	Definizione di usabilità.....	92
4.1.2	Valutazione di usabilità	92
4.1.3	Metodi e approcci per la valutazione dell'usabilità	93
4.1.4	Attributi di usabilità.....	97
4.1.5	Metriche di valutazione	98
4.2	Preparazione dell'esperimento di usabilità sugli ILs	99
4.2.1	Esperimento preliminare: test di comprensione	99
4.2.2	Generalità, obiettivi	101
4.2.3	Metodo di valutazione	101
4.2.4	Scelta degli utenti e loro distribuzione nell'esperimento.....	109
4.2.5	Descrizione delle due modalità di fruizione	111

4.3	Risultati dell'esperimento	113
4.3.1	Apprendimento della tecnica di interazione con il mouse.....	113
4.3.2	Apprendimento della tecnica di interazione con il joystick	115
4.3.3	Apprendimento della tecnica di interazione: confronto tra mouse e joystick	117
4.3.4	Test di navigazione (travel) su PC con mouse	118
4.3.5	Test di navigazione (travel) nel CAVE con joystick.....	119
4.3.6	Test di navigazione: confronto tra mouse e joystick.....	120
4.3.7	Test di orientamento	121
4.3.8	Soddisfazione dell'utente	128
4.4	Conclusioni sul test	136
4.4.1	Riepilogo delle osservazione fatte durante il test	136
4.4.2	Riepilogo dei maggiori problemi di usabilità in modalità desktop	137
4.4.3	Riepilogo dei problemi di usabilità in modalità immersiva	138
4.4.4	Possibili soluzioni ai problemi di usabilità nella modalità desktop.....	138
4.4.5	Possibili soluzioni ai problemi di usabilità nella modalità immersiva	142
4.4.6	Possibili linee guida per la progettazione di IL	143
	Conclusioni e sviluppi futuri.....	145
	APPENDICE A (Elenco dei comandi)	147
	APPENDICE B (Discorso introduttivo all'esperimento).....	156
	APPENDICE C (Modulo per il consenso informato).....	157
	APPENDICE D (Questionario finale dei test).....	158
	Bibliografia	164

Introduzione

Dato il ruolo pervasivo che lo spazio gioca nelle attività umane, le attività di percezione, orientamento e movimento nello spazio sono in gran parte attività subcoscienti che richiedono un basso carico cognitivo, mentre l'apprendimento e la memorizzazione di grandi quantità di informazioni sono attività fortemente impegnative e faticose.

Per questo motivo, fin dall'antichità si è fatto ricorso a metodi per facilitare l'organizzazione, l'apprendimento e la memorizzazione di informazioni che sfruttassero la nostra abilità di trattare con lo spazio. Le tecniche di memorizzazione basate sull'associazione di informazioni in locazioni spaziali risalgono almeno al 500 a.C. Le informazioni, organizzate spazialmente diventano più facili da comprendere, memorizzare, utilizzare come punto di inizio nella generazione di nuove idee.

In quest'ottica, si collocano i Paesaggi di Informazione (Information Landscapes), particolari tipi di applicazione di realtà virtuale che anziché proporre ambienti virtuali popolati di elementi realistici o verosimili, mettono in scena elementi astratti come testi, immagini (o al più modelli 3D) disposti nell'ambiente tridimensionale, in cui l'utente può muoversi liberamente o seguendo percorsi predefiniti.

Essi forniscono uno strumento in grado di organizzare spazialmente le informazioni, mettendone in evidenza possibili strutture e relazioni, ma senza mai imporsi rigidamente all'utente che viene lasciato libero nel suo percorso di conoscenza, con possibili benefici nell'ambito della comunicazione e nell'apprendimento.

Inoltre, rispetto ad altri mezzi di comunicazione digitali come gli ipertesti, i Paesaggi di Informazione potrebbero essere considerati come una evoluzione, dato che permettono di mantenere sempre una visione del contesto in cui si colloca una specifica informazione, e permettono inoltre di raggiungere una vera e propria "non sequenzialità" piuttosto che una "multi-sequenzialità".

Tuttavia nonostante i vantaggi che teoricamente sembrerebbero fornire, nella realtà gli Information Landscapes non sono stati comunemente e ampiamente adottati. Le cause possono essere molteplici. Da una parte, il fatto che non siano stati sufficientemente diffusi sistemi che permettano a chiunque di creare i propri Paesaggi di Informazione e di condividerli; dall'altra, una non accettazione di questa nuova modalità di accesso

all'informazione da parte degli utenti, dovuta alla scarsa usabilità dei sistemi di visualizzazione e interazione.

Questo lavoro di tesi si è posto lo scopo di colmare almeno in parte queste lacune a partire dalla soluzione software adottata presso il Lab. PERCRO della Scuola Superiore S.Anna di Pisa. In particolare, la tesi si focalizza su vari aspetti:

- progettazione e sviluppo di un editor web per la creazione di Paesaggi di Informazione;
- sviluppo di nuove funzionalità che possano agevolare l'utilizzo dei Paesaggi di Informazione da parte dell'utente, come per esempio la funzionalità di ricerca testuale;
- avvio di uno studio di usabilità, per mezzo di test con utenti, allo scopo di identificare le maggiori problematiche, ipotizzarne soluzioni e, in alcuni casi, implementarle.

Riteniamo che una maggiore attenzione all'usabilità dei sistemi di visualizzazione e delle tecniche di interazione utilizzate con gli ILs, così come lo sviluppo di funzionalità di editor accessibili da web, possano contribuire ad una maggiore accettazione di questa nuova modalità di fruizione dell'informazione e alla sua diffusione, o almeno produrre nuova conoscenza nell'ambito di questi sistemi e delle problematiche ad essi connesse.

Il documento di tesi è così strutturato:

- nel *capitolo 1* si fornisce un inquadramento teorico e una definizione dell'argomento trattato descrivendone lo stato dell'arte;
- il *capitolo 2* è dedicato all'analisi del software di visualizzazione XVR per Information Landscapes sviluppato presso il Lab. PERCRO;
- il *capitolo 3* è dedicato alla descrizione dello strumento di editing sviluppato, e alle scelte progettuali e realizzative. Sempre in questo capitolo si descrive la funzionalità di ricerca testuale sviluppata;
- il *capitolo 4* è dedicato all'usabilità. Dopo un'introduzione teorica, si descrive l'esperimento di usabilità realizzato e si riportano i risultati. Si fornisce inoltre una descrizione dei principali problemi di usabilità rilevati, proponendo soluzioni. Di alcune di queste viene proposta un'implementazione, come nel caso della mappa 2D per Information Landscapes.

- la tesi termina illustrando le *conclusioni* sul lavoro svolto e sui possibili sviluppi futuri.

1 Stato dell'arte

Oggetto di questa tesi sono i *Paesaggi di Informazione*, o in termini anglosassoni gli *Information Landscapes* (ILs). Secondo una definizione molto generale, essi consistono in ambienti virtuali interattivi che presentano al fruitore informazioni astratte (testi e immagini) strutturate in uno spazio tridimensionale, esplorabile liberamente o seguendo percorsi predefiniti dal progettista (designer) dell'ambiente (Figura 1).

In questo primo capitolo, oltre a fornire un inquadramento teorico e una definizione più puntuale dell'argomento trattato, si cercherà di fornirne uno stato dell'arte, sia dal punto di vista delle applicazioni pratiche realizzate che dello studio e della ricerca che sono stati portati avanti in questo ambito.

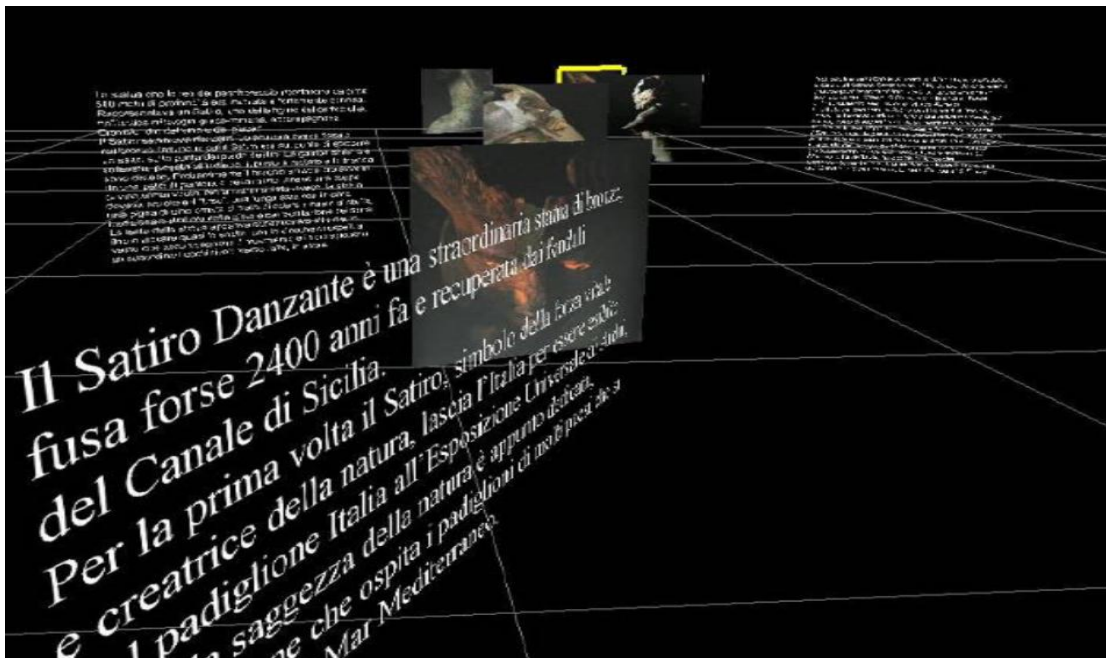


Figura 1. Esempio di information landscape (immagine tratta dall'IL "IL SATIRO", PERCRO LAB)

1.1 Per un inquadramento dell'argomento

Per loro natura, i Paesaggi di Informazione possono essere collocati al confine tra due discipline distinte: la *Realtà Virtuale* e la *Visualizzazione dell'Informazione*. Se considerato dal punto di vista della realtà virtuale, un Information Landscape può essere visto come un caso particolare di ambiente virtuale, un ambiente astratto che anziché essere popolato di oggetti realistici o verosimili, presenta al fruitore informazioni astratte come testi o immagini. Considerato dal punto di vista della *Information Visualization*, l'IL può essere pensato come una strategia di visualizzazione di informazione che fa uso degli strumenti e dei metodi della realtà virtuale *come mezzo* per presentare l'informazione. Un'altra disciplina di confine è il graphic design e in particolare quel filone che si occupa dello studio della tipografia virtuale, cioè della comunicazione tipografica attraverso i media digitali. In questa sezione cercheremo di approfondire quanto appena esposto.

1.1.1 Gli Information Landscapes come caso specifico di realtà virtuale

Per poter dare una collocazione ai Paesaggi di Informazione nell'ambito della disciplina (o meglio meta-disciplina) della realtà virtuale, occorre avere almeno un'idea di che cosa essa si occupi, e quali tipi di realtà virtuale esistono.

Il termine realtà virtuale nasce nel 1988 durante un'intervista a Jaron Lanier, informatico, compositore e saggista statunitense, considerato un pioniere della realtà virtuale (Lanier, 1989). Il termine è difficilmente definibile anche se tutte le definizioni che sono state tentate condividono alcune parole chiave come: ambiente, computer, sensi, utente, simulazione, interazione, immersione, presenza. La realtà virtuale, infatti, propone un ambiente 3D (che non necessariamente ha legami con la realtà e le sue regole), generato artificialmente da un computer. All'interno di tale spazio, l'utente può muoversi e vedere gli oggetti da vari punti di vista o interagire con essi per mezzo di apposite interfacce.

L'utilizzo di interfacce che coinvolgono i vari sensi umani, permette il raggiungimento di un certo livello di immersività e di presenza. L'immersività si riferisce alla sensazione fisica di trovarsi in un altro mondo: essa è molto legata alla

disponibilità di determinate tecnologie capaci di isolare dal mondo reale e sollecitare i vari sensi con stimoli vari e in modo credibile; la presenza è invece la sensazione mentale di trovarsi in uno spazio virtuale (nozione applicabile anche a media tradizionali come libri o film).

A seconda di come vengono progettati, dei dispositivi di interazione e di visualizzazione adottati, possiamo avere sistemi di realtà virtuale molto diversi tra loro. Nel tempo sono stati proposti diversi metodi di classificazione, ciascuno dei quali prende in considerazione un determinato aspetto caratterizzante (Carrozzino et al., 2010).

Dal punto di vista dell'immersività possiamo distinguere tra realtà virtuale non immersiva (quella che utilizza principalmente dispositivi desktop per la visualizzazione e l'interazione), VR poco immersiva (quella che fa uso di dispositivi indossabili come caschetti, cuffie e altro), e infine VR altamente immersiva (quella che utilizza dispositivi esterni come CAVE, motion platform ecc). (Figura 2)

Immersion

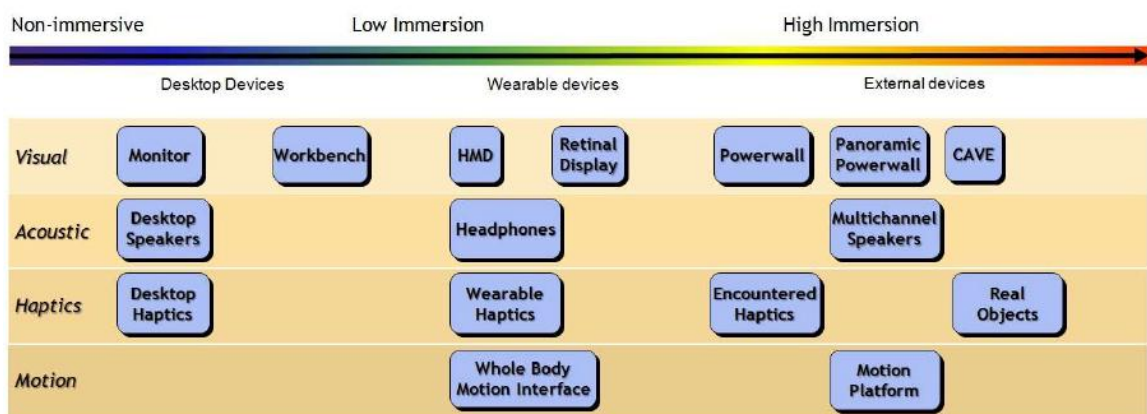


Figura 2. Classificazione della VR in base all'immersione

Dal punto di vista dell'interazione possiamo distinguere tra sistemi non interattivi, i quali non permettono alcun tipo di interazione con l'ambiente, sistemi che si basano sull'interazione mediata da dispositivi e sistemi caratterizzati da una interazione naturale con l'ambiente virtuale ad esempio per mezzo di gesti riconosciuti direttamente dal sistema o il linguaggio). (Figura 3)

Interaction

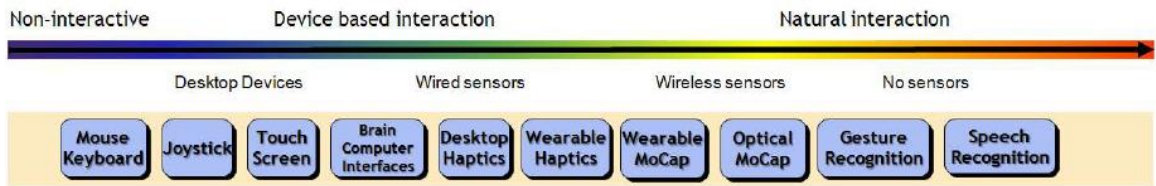


Figura 3 classificazione della VR in base all'interazione

Infine, se consideriamo l'aspetto dell'astrazione dalla realtà, possiamo identificare un continuum che va dalla riproposizione il più possibile dettagliata della realtà come essa è utilizzando a tal scopo dispositivi di campionamento come gli scanner 3D per l'acquisizione di ambienti e oggetti, alla proposizione di un ambiente reale ma tramite modellazione (sintesi), alla modellazione di un ambiente non reale ma realistico (plausibile nella realtà), fino ad arrivare alla proposizione di ambienti parzialmente astratti o totalmente astratti e avulsi dalla realtà, cioè senza elementi di aggancio con essa (Figura 4).

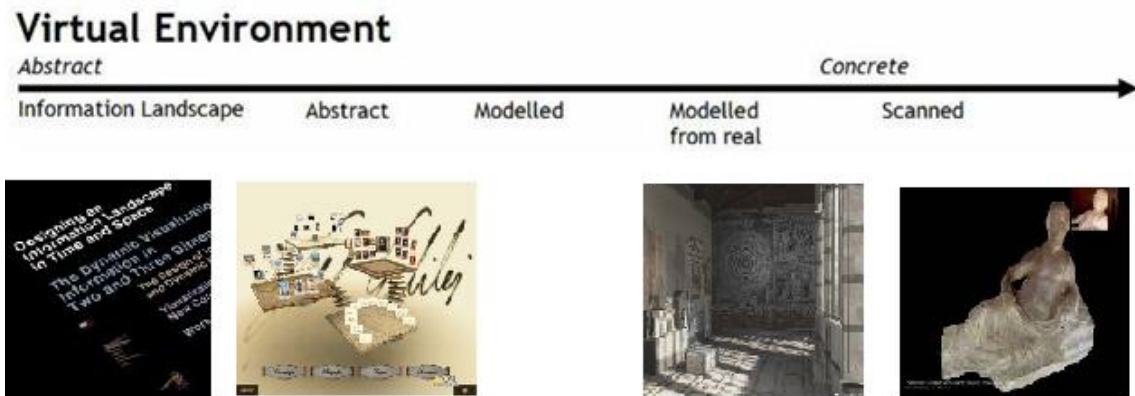


Figura 4 Classificazione della realtà virtuale in base all'astrazione

Nell'ambito degli ambienti parzialmente astratti possiamo far rientrare quegli ambienti poco plausibili perché modellano una realtà alternativa o perché possiedono una forte presenza di informazioni astratte come testi, immagini e video (Polys et al., 2004) che si integrano e arricchiscono l'ambiente vero e proprio.

Nel contesto degli ambienti totalmente astratti, si collocano gli **Information Landscapes**, particolari tipi di ambienti virtuali che non hanno una corrispondenza con una controparte reale ma sfruttano lo spazio tridimensionale per mappare le relazioni semantiche tra dati di natura testuale in relazioni spaziali. Bisogna precisare che se ai dati testuali aggiungiamo anche dati di diverso tipo (come video, foto, musica) otteniamo qualcosa di più del semplice Information Landscape (IL), che possiamo definire come Multimedia Information Landscape (MIL). In riferimento alla dimensione immersiva, sono stati sviluppati sistemi di IL sia per la fruizione desktop, sia per la fruizione immersiva. Dal punto di vista dell'interazione, gli IL così come sono stati sviluppati finora, sono da considerare come sistemi di realtà virtuale basati sull'interazione tramite dispositivi, i quali consentono all'utente l'esplorazione dell'ambiente informativo.

1.1.2 I paesaggi di informazione nell'ambito dell'Information Visualization

Seguendo l'approccio della disciplina denominata realtà virtuale abbiamo definito gli IL come ambienti virtuali costituiti principalmente di informazioni astratte.

Ma gli IL, sostanzialmente, non sono altro che sistemi per la visualizzazione di informazioni, per cui possiamo chiamare in causa un'altra disciplina specifica, l'Information Visualization, la quale pone al centro dei suoi interessi proprio la rappresentazione grafica delle informazioni. L'espressione *Information Visualization* è stata coniata alla fine degli anni '80 da alcuni ricercatori appartenenti allo User Interface Research Group della Xerox PARC per distinguere una nuova disciplina che si occupava dello studio e della creazione di artefatti visivi allo scopo di amplificare la cognizione, esigenza che si faceva sempre più pressante con l'esplosione di dati dovuta alla globalizzazione dell'economia e della comunicazione, e soprattutto al rapido avanzare della tecnologia (Mazza, 2009).

La rappresentazione visiva del dato ha il vantaggio di utilizzare alcune proprietà grafiche che vengono processate molto velocemente ed efficientemente dalla percezione visiva: gli attributi visivi su cui i dati vengono mappati (come il colore, la dimensione, la prossimità o il movimento) sono infatti immediatamente catturati e processati dal sistema percettivo visivo, perfino prima dell'entrata in gioco dei processi cognitivi della mente. È proprio per questo motivo che le rappresentazioni

favoriscono la comprensione di insiemi di dati complessi, supportano le decisioni, permettono di estrapolare informazioni che altrimenti rimarrebbero nascoste nei dati. La parte difficile sta quindi nel progettare delle rappresentazioni visive che riescano nell'obiettivo per cui sono state create. A tale scopo, alcune indicazioni ci vengono date da Edward Tufte, uno dei principali studiosi in questo campo, il quale afferma che l'eccellenza si raggiunge quando si riescono a comunicare idee complesse con "chiarezza, precisione ed efficienza" (Tufte, 1983). Egli riassume inoltre alcuni principi per un corretto design: a) mostrare i dati; b) aiutare il fruitore a concentrarsi sull'informazione piuttosto che sul design; c) favorire il confronto dei dati; d) rendere coerenti grandi insiemi di dati. Per ulteriori informazioni sull'argomento si vedano i numerosi testi di Tufte (1983, 1990, 1997, 2006). Inoltre, una profonda comprensione del funzionamento dei meccanismi percettivi umani non può che favorire la creazione di rappresentazioni grafiche efficaci (Ware, 2004).

Le tecnologie dell'informazione costituiscono uno strumento utile sia per la creazione che per la proposizione di rappresentazioni grafiche più efficaci, in quanto i computer permettono di trattare e compiere operazioni su grandi moli di dati difficilmente gestibili altrimenti, e permettono inoltre di introdurre l'elemento dell'interattività, consentendo all'utente di compiere azioni come trasformazioni della rappresentazione, filtraggio di dati in input, cambio del tipo di rappresentazione (Ware, 2004; Mazza, 2009).

Secondo l'approccio dell'Information Visualization, gli ILs possono quindi essere pensati come rappresentazioni visuali di dati astratti (testi ed immagini), che utilizzano una particolare tecnologia ovvero la realtà virtuale come mezzo di comunicazione. Come è insito nei sistemi di realtà virtuale, i dati saranno visualizzati in uno spazio tridimensionale secondo la metafora del paesaggio, e sarà possibile esplorarli attraverso una qualche forma di interattività.

È importante notare che gli IL non costituiscono un caso unico di utilizzo del 3D nella visualizzazione dell'informazione. La disciplina dell'*information visualization* ha accettato ormai da anni l'uso della terza dimensione dato che permette di sfruttare per la rappresentazione alcuni elementi non presenti o limitati nella visualizzazione bidimensionale come: profondità, movimento, distanza, trasparenza, animazione e orientamento spaziale. Parecchie sono le tecniche esistenti per la rappresentazione tridimensionale dei dati: esse differiscono tra loro per le metafore di visualizzazione e di navigazione, e per i tipi di dati rappresentati. Tra queste, meritano sicuramente di

essere menzionate le seguenti: perspective wall (Robertson et al., 1991), Cone Trees (Card, 1991), Web Book (Card, 1996), grafi generici (Figura 5).

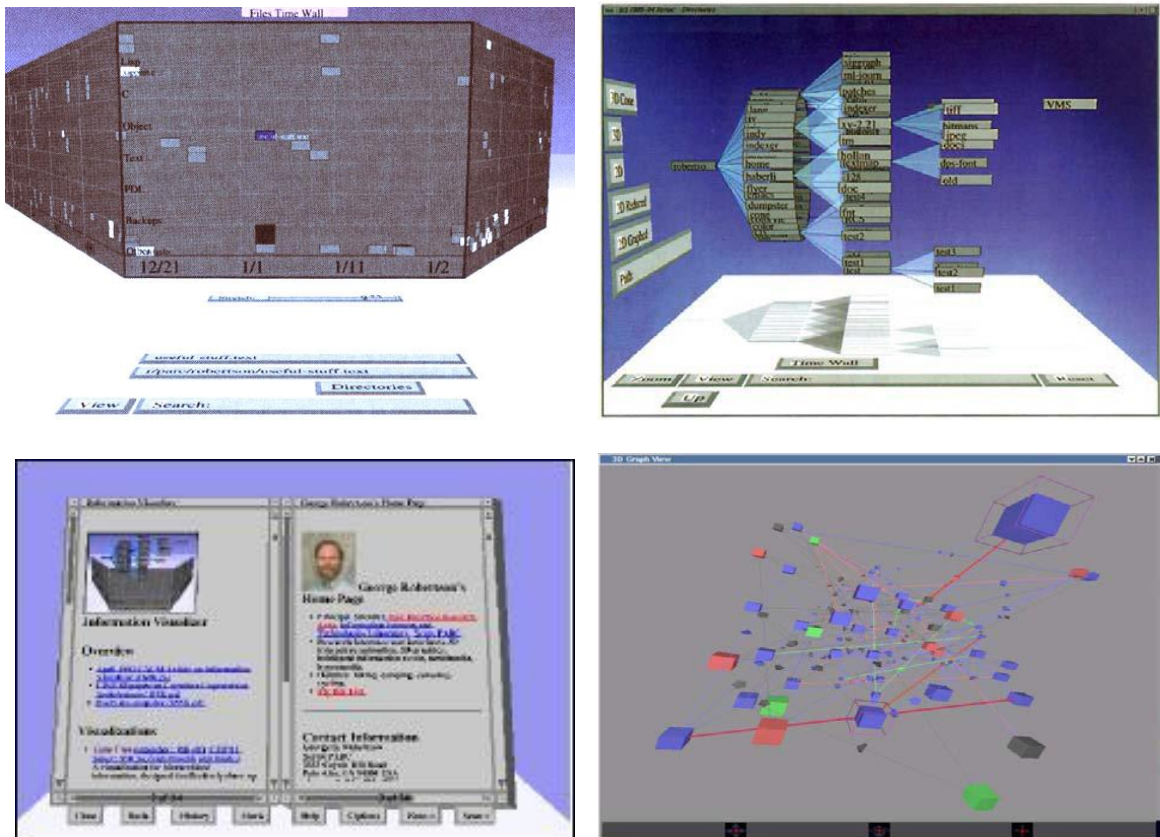


Figura 5 Perspective wall (in alto a sinistra), Cone Trees (in alto a destra), Web Book (in basso a sinistra), grafo generico (in basso a destra)

Nonostante i vantaggi che la rappresentazione 3D comporta soprattutto quando si vogliono rappresentare grandi quantità di dati in modo efficace, c'è chi sostiene che occorra limitarne l'utilizzo alle situazioni in cui sia veramente necessaria. Studi empirici infatti hanno mostrato come in genere la rappresentazione tridimensionale incrementi il carico cognitivo e lo sforzo mentale dell'utente per interpretare correttamente il dato rappresentato (Mazza, 2009).

In conclusione, possiamo affermare che il tema dei paesaggi di informazione sia un argomento di confine che sfrutta (o dovrebbe sfruttare) le potenzialità di entrambe le discipline.

La realtà virtuale eccelle nel fornire un mezzo per la rappresentazione grafica nello spazio di oggetti con modalità più o meno immersive e interattive; l'Information Visualization fornisce metodi per migliorare come l'utente percepisce, apprende e interagisce con rappresentazioni visive di informazioni astratte.

1.1.3 Graphic Design e ILs: La tipografia virtuale

Una disciplina che si è interessata all'argomento, è quella denominata graphic design. Il motivo sta nel fatto che, come avremo modo di vedere nel seguito, la creazione di un Paesaggio di Informazione include un'intensa attività preliminare di progettazione, ma anche perché l'utilizzo della terza dimensione nella visualizzazione dei testi pone nuove problematiche interessanti nell'ambito della tipografia virtuale. Non va dimenticato infatti che gli Information Landscape nascono prima di tutto dalla curiosità negli anni novanta di esplorare nuove possibilità nell'ambito del design tipografico da parte di una studiosa, Muriel Cooper, appartenente al MIT Media Laboratory di Cambridge. Sperimentando attraverso l'uso di un software denominato Typographic Space che verrà descritto nel paragrafo che segue, emersero una prima serie di osservazioni. Il primo problema della tipografia tridimensionale è la distorsione del testo causata sia dalla prospettiva che dalla possibilità di scegliere un punto di vista arbitrario. Per esempio, il punto di vista può essere collocato anche dietro il testo con la conseguenza di avere una visione speculare del testo; il testo diviene invece una linea o scompare se osservato da un lato. Un modo per risolvere il problema è mostrare sempre il testo frontalmente all'utente. Un secondo problema della tipografia tridimensionale è data dal fatto che la dimensione percepita del testo cambia al variare della posizione dell'utente nello spazio, per cui la grandezza del font non può essere usata, così come succede nel design tradizionale, come indizio visuale per indicare un certo tipo di informazione. Infatti, un testo con font piccolo localizzato vicino al viewpoint può avere la stessa dimensione percepita di un testo di dimensioni più grandi ma più lontano. Tuttavia, quando c'è una chiara struttura visuale nello spazio, anche la dimensione può essere usata in maniera efficace. Una terza osservazione consiste nel fatto che il movimento tridimensionale può essere usato come nuova forma di espressione nella tipografia, estendendo la complessità del messaggio che il designer può creare rispetto alle animazioni 2D (Small et al., 1994).

Nell'ambito della tipografia virtuale, sono stati in seguito effettuati nuovi studi. Per quanto riguarda l'effetto dell'orientamento sulla leggibilità, è stato dimostrato come la rotazione del testo sull'asse x determini un aumento del tempo di lettura (Larson et al., 2000). Una generalizzazione a tutte le possibili rotazioni è stata effettuata da Grossman et al. (2007) insieme ad una quantificazione precisa di tale effetto. Sempre in Grossman et al. (2007) si è scoperto come la rotazione di determinate parole può determinare ambiguità visive che possono riguardare sia i singoli caratteri che le parole, impedendo all'utente di riconoscere correttamente il testo, e sono state proposte tecniche di

disambiguazione. Per questi motivi coloro che hanno studiato l'integrazione di informazione testuale all'interno di ambienti di realtà virtuale, hanno dimostrato tramite esperimenti con utenti come sia preferibile visualizzare il testo anziché direttamente all'interno dell'ambiente tridimensionale (*within-the-world technique*) su un piano 2D sovrapposto alla scena 3D (*Heads-up Display Technique*) (Chen et al., 2004). Bisogna precisare però che questo modo di operare funziona bene nel caso in cui lo scopo sia quello di costruire un IRVE (*Information Rich Virtual Environment*), in cui l'informazione astratta (testo, audio e video) fa da complemento all'ambiente virtuale vero e proprio. Non è possibile utilizzarla invece nel caso degli IL dove l'informazione è l'unica protagonista dell'ambiente.

Un altro aspetto che è stato studiato riguarda il grado di leggibilità di vari stili di disegno del testo su sfondi diversi. In un esperimento (Jankowski, 2010) sono stati investigati gli effetti sulla leggibilità del testo al variare di: a) stile di disegno del testo (normale, sopra un billboard trasparente, usando una tecnica denominata *Anti-interference*, usando un ombreggiatura), b) tipo di presentazione (in positivo ovvero testo nero su billboard bianco o con ombreggiatura bianca, o in negativo ovvero testo bianco su billboard nero o con ombreggiatura nera), c) elemento in background (video o ambiente 3D). È stato dimostrato come: a) ci sia poca differenza in performance di lettura passando dal background video a quello 3D; b) la presentazione in negativo è migliore di quella in positivo; c) lo stile di disegno con billboard produce migliori performance ed è preferito dagli utenti. Benché tale esperimento sia stato fatto renderizzando un testo su un piano 2D sovrapposto al video o all'ambiente 3D, riteniamo che tali risultati possano essere estesi anche nell'ambito dei testi renderizzati dentro un ambiente 3D: un testo in 3D ha infatti anch'esso la problematica della sua riconoscibilità rispetto allo sfondo che consiste solitamente in modelli tridimensionali.

Infine, è stato approfondito l'effetto dell'animazione del testo sulla sua leggibilità. Nell'ambito della *motion graphics* è nato un vero e proprio campo denominato *kinethic typography*, il quale utilizza l'animazione per illustrare dei concetti. Gli elementi aggiuntivi del tempo e del movimento forniscono opportunità che arricchiscono la comunicazione tipografica tradizionale. Tuttavia, ci sono studi che affermano un aumento dei tempi di lettura come conseguenza dell'uso dell'animazione nel testo (Ditterline et al., 2000) così come una diminuzione della leggibilità (Minakuchi, 2008).

1.2 Applicazioni esistenti

In questo paragrafo cercheremo di riportare alcuni esempi di applicazioni di grafica tridimensionale interattiva che utilizzano la metafora del paesaggio per la visualizzazione delle informazioni. Di essi riporteremo una descrizione delle caratteristiche, che sarà usata nei paragrafi successivi come base di partenza per operare una classificazione di questi sistemi e individuare gli ambiti in cui sono stati finora utilizzati.

1.2.1 Typographic Space (MIT)

Il software denominato "Typographic Space" è stato realizzato nel 1994 presso il MIT *Media Laboratory* di Cambridge e rappresenta uno dei primissimi esempi di software per la gestione di Paesaggi di Informatione (Small et al.,1994). È proprio in questo ambito che Muriel Cooper conia l'espressione "Information Landscape" ad intendere un ambiente virtuale dove le informazioni testuali sono posizionate su superfici collocate in uno spazio 3D e in cui l'utente può navigare mantenendo nel contempo una piena percezione del contesto .

Designer di formazione, nel 1988 Cooper aveva formato il *Visibile Language Workshop*, un laboratorio dove si investigavano questioni relative alla tecnologia e il design. Insieme agli studenti Suguru Ishizaki e David Small, Cooper si era posta l'obiettivo di applicare le tecniche tipografiche sviluppate nell'ambito del design grafico tradizionale bidimensionale in ambito tridimensionale, e investigare i problemi nell'uso della grafica tridimensionale interattiva come mezzo di comunicazione tipografica.

Il software, quindi, consente la visualizzazione di testi collocati in uno spazio 3D navigabile. Tra le funzionalità fornite dal tool, vi è anche la possibilità di cambiare un insieme di attributi tipografici allo scopo di sperimentare vari aspetti della tipografia tridimensionale. Per l'interazione esso fornisce una semplice interfaccia costituita da mouse e tastiera: il mouse è usato per cambiare la *view distance*, mentre un insieme di otto tasti sono assegnati al settaggio della rotazione, e traslazione del punto di vista. Purtroppo, gli articoli non forniscono dettagli tecnici circa l'implementazione del software. Per la presentazione del lavoro insieme all'articolo fu creato un video disponibile su youtube¹ (Figura 6).

¹ <http://www.youtube.com/watch?v=Qn9zCrlJzLs>

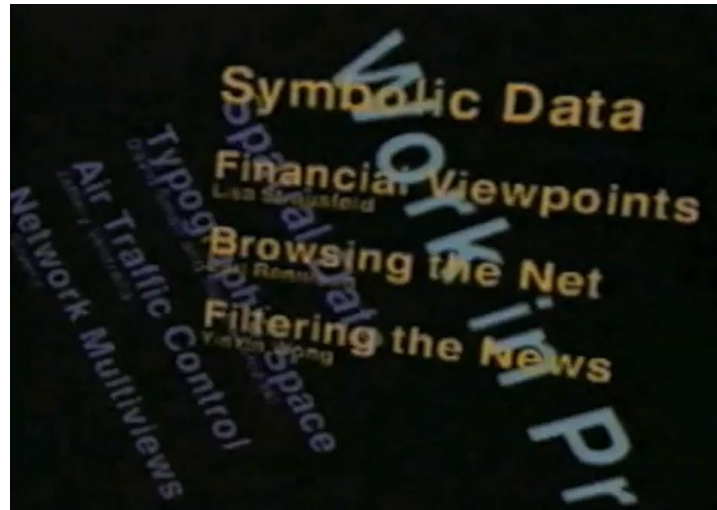


Figura 6. Immagine tratta dal video di presentazione del software typographic space

Poco dopo la realizzazione di questo lavoro, Muriel Cooper morì, ma il lavoro iniziato venne portato avanti dall'allievo David Small con la presentazione di un progetto in cui il software Typographic Space veniva utilizzato per la visualizzazione di un paesaggio di informazione specifico (Small, 2004). Il lavoro, denominato *Virtual Shakespeare* prevedeva la strutturazione di tutte le opere di W. Shakespeare sottoforma di Paesaggio di Informazione. La scelta è ricaduta su queste opere sia per la grande quantità di informazione da gestire (circa un milione di parole) sia per la strutturazione che le caratterizza (organizzazione in dialoghi, scene, atti ecc). Per ciascuna opera, gli atti sono stati visualizzati in colonne ciascuna delle quali è suddivisa in scene. Si fa inoltre un ampio utilizzo dei livelli di dettaglio: quando l'utente si trova vicino al testo questo risulta leggibile, ma man mano che si allontana la visualizzazione mostra le scene, gli atti, fino a mostrare tutta l'opera ma perdendo il dettaglio sullo scritto (Figura 7). Le varie parti che compongono l'opera (ad esempio i dialoghi dei diversi personaggi) sono differenziate attraverso l'uso di font e colori differenti che permettono di percepire con uno sguardo informazioni di diversa natura (per esempio l'importanza di un personaggio all'interno di una scena).



Figura 7. Quattro viste dell'IL virtual shakespeare secondo livelli di dettaglio differenti

Nel realizzare questo progetto D. Small ha ripensato anche la modalità di interazione: anziché utilizzare tastiera e mouse, poco adatti per l'utilizzo in ambiente tridimensionale, l'autore ha proposto tre soluzioni originali che utilizzano strumenti realizzati con l'uso dei mattoncini LEGO e di sensori di campo magnetico per determinare l'esatta posizione e orientazione dell'oggetto nello spazio.

La prima soluzione utilizza due sensori da tenere uno nella mano destra, l'altro nella sinistra. Un sensore serve per controllare la posizione e orientazione del testo e agisce come se fosse una maniglia; il secondo, incorporato in un piccolo elicottero LEGO serve a controllare attraverso il movimento la posizione e orientazione della camera nello spazio.

La seconda soluzione utilizza una piccola costruzione LEGO rappresentante un palcoscenico. Due rotelle per il posizionamento sono collocate nella struttura in due posizioni tra loro ortogonali. Per spostare il testo in alto e in basso viene utilizzata la rotella collocata su un lato del palcoscenico, mentre per spostare il testo a destra e a sinistra, si utilizza quella che si trova sopra il palcoscenico. L'angolo di vista del testo è controllato ruotando l'intero palcoscenico. È infine possibile evidenziare il testo relativo ad un determinato personaggio collocando fisicamente un diverso attore LEGO sul

palcoscenico. Questo viene riconosciuto grazie ad una resistenza saldata ai piedi di ogni attore.

La terza soluzione consiste in un dispositivo portatile costruito con legno e costruzioni LEGO e che unisce il display con i controlli di navigazione. Il prototipo presenta un solo bottone sulla destra: quando questo è premuto, il movimento del display ha l'effetto di muovere la camera virtuale all'interno dell'information landscape. La velocità del movimento di camera cresce esponenzialmente con la quantità di movimento del display, offrendo all'utente sia un controllo fine su piccoli movimenti, sia la possibilità di movimenti veloci e ampi. Se il bottone viene rilasciato la visualizzazione rimane bloccata.

1.2.2 Lab. Percro (Scuola superiore S. Anna)

Il laboratorio di Robotica Percettiva (PERCRO) della Scuola Superiore S. Anna di Pisa, si è interessato da tempo allo sviluppo di sistemi per la creazione e visualizzazione dei Paesaggi di Informazione (Ruffaldi et al., 2008; Carrozzino et al., 2009). La ricerca è iniziata con lo sviluppo di due applicazioni per piattaforma Windows: un viewer per la visualizzazione di ILs e un editor per facilitare la creazione di un Paesaggio di Informazione (Figura 8).

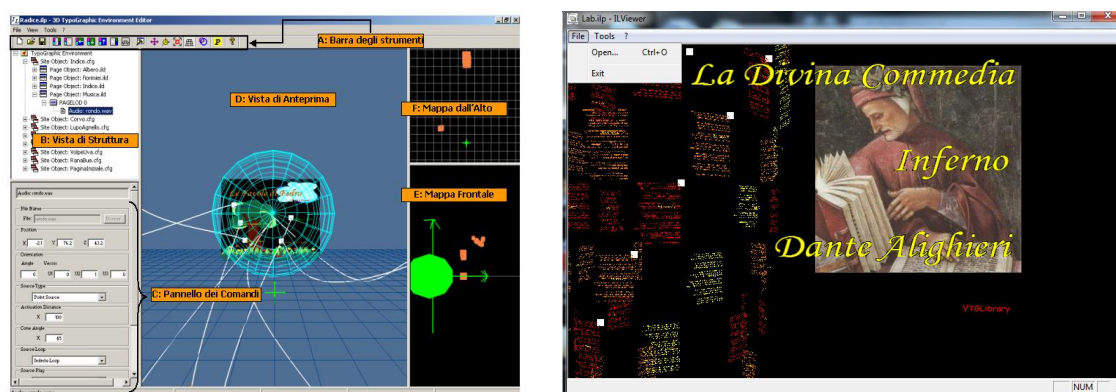


Figura 8 ILEditor (a sinistra), ilviewer (a destra). Lab. PERCRO.

L'editor IL permette la creazione e gestione di Paesaggi di Informazione. Esso presenta caratteristiche simili ai programmi di modellazione 3D, anche se orientato a gestire particolari strutture di dati gerarchiche che servono a descrivere un paesaggio di Informazione.

I dati che costituiscono un information landscape possono essere immaginati come organizzati secondo una struttura gerarchica ad albero (Figura 9), ai cui nodi sono associate entità denominate parchi, siti, pagine, e foglie. Le foglie rappresentano i contenuti informativi veri e propri (testi, immagini, modelli 3D); più elementi informativi possono essere raggruppati in una singola pagina; più pagine che presentano affinità semantiche possono essere raggruppate in un contenitore più grande denominato sito; più siti affini possono essere raggruppati in un Parco.

Ogni entità presenta il proprio sistema di riferimento, e le entità di più basso livello sono posizionate rispetto al sistema di riferimento del padre. È possibile inoltre specificare che le pagine assumano un aspetto diverso al variare della distanza di osservazione, attraverso l'uso dei livelli di dettaglio (LOD), entità speciali che possono essere poste come figli delle entità pagina e che conterranno gli elementi da gestire quando diventano attivi. La distanza di attivazione è quindi una proprietà che deve essere specificata per un'entità LOD. L'uso dei LODs consente per esempio di visualizzare solo il titolo di una pagina quando questa è osservata da una distanza elevata, alcuni sottotitoli a distanza minore, il contenuto completo quando l'utente è molto vicino.

È possibile inoltre definire dei collegamenti (link) tra foglie i quali verranno visualizzati come linee nello spazio: in questo contesto il concetto di link si trasforma rispetto al web dato che il link può essere pensato come un viaggio nello spazio, un percorso che guida l'utente da un concetto all'altro mantenendo una visione su tutto lo spazio. Per questo motivo un link può collegare più di due elementi a formare un vero e proprio percorso.

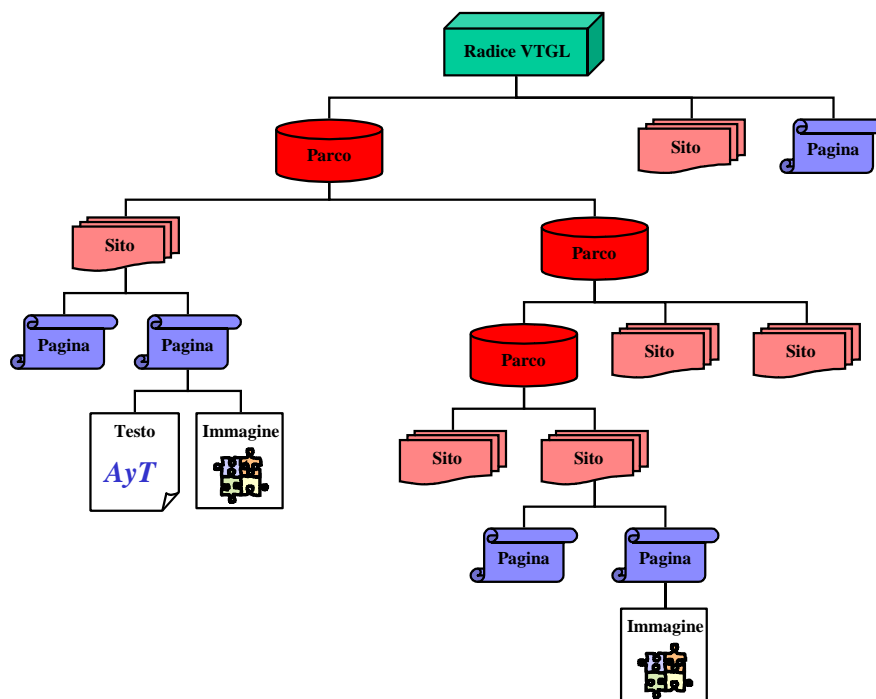


Figura 9 Organizzazione logica di un paesaggio di informazione (PERCRO)

L'editor facilita la creazione della struttura organizzativa dell'IL, l'inserimento di contenuti (testi, immagini, modelli 3D in formato VRML), il loro posizionamento, orientamento e scala, l'impostazione di parametri di formattazione (es. come colore, fonts e animazioni per il testo, ecc.).

L'editor mette inoltre a disposizione una modalità di preview che permette di verificare la struttura dell'IL creato, le proprietà e i comportamenti.

La struttura e le proprietà specificate nell'editor vengono organizzati e memorizzati fisicamente in file (Figura 10). In particolare la composizione della pagina è descritta in un file con estensione .ILD in cui vengono specificati i diversi livelli di dettaglio previsti, i riferimenti agli oggetti da visualizzare, i relativi attributi di visualizzazione, eventuali link verso altri oggetti. La struttura del sito è specificata in un file di configurazione con estensione .CFG in cui sono contenute le informazioni relative all'organizzazione delle pagine nel sito. Analogamente, la struttura di un parco (a tutti gli effetti un sito di siti) è specificata attraverso un file .CFG. Complessivamente, il paesaggio di informazione viene descritto in un file con estensione .ILP in cui oltre alla definizione degli oggetti presenti al primo livello dell'albero, sono specificate le proprietà globali dell'ambiente, come il colore di sfondo, le caratteristiche della griglia, la posizione iniziale di camera.

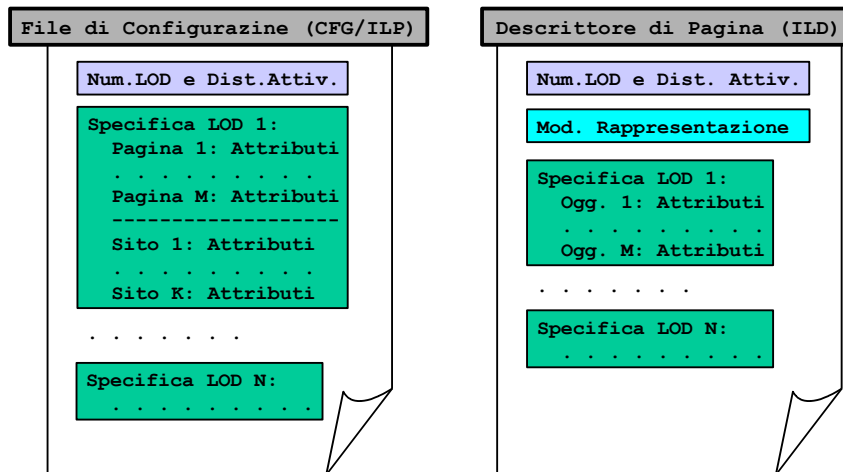


Figura 10 L'organizzazione in files dei paesaggi di informazione (PERCRO)

Il Paesaggio di informazione creato tramite l'editor e la cui struttura e caratteristiche sono memorizzate in file, può essere caricato e visualizzato in un visualizzatore, un'applicazione utilizzabile in ambiente Windows. Una volta avviata l'applicazione, occorre selezionare il dispositivo che si intende utilizzare per l'interazione. Essa può avvenire attraverso gli strumenti classici come la tastiera e il mouse o per mezzo di dispositivi più adatti per la navigazione in ambienti tridimensionali come il joystick, lo space mouse (particolare tipo di mouse dotato di sei gradi di libertà) e il cyberman. Lo svantaggio principale di tale visualizzatore è la mancata possibilità di essere integrato all'interno di una pagina web, rendendo di fatto impossibile questo tipo di fruizione dei paesaggi di informazione.

In anni più recenti è stata realizzata una nuova versione dell'engine per la visualizzazione dei paesaggi di informazione in modo da renderlo eseguibile all'interno dell'ambiente XVR (Carrozzino et al., 2005), un framework tecnologico sviluppato presso il medesimo laboratorio e utilizzato ormai da anni per gestire applicazioni interattive e multimediali di realtà virtuale, sia in ambiente desktop o immersivo, sia in ambiente web (come plug-in all'interno del browser Internet Explorer) .

Dato che questo lavoro di tesi si propone di portare avanti ciò che è già stato fatto presso il Lab. PERCRO, utilizzando come base l'engine XVR, maggiori dettagli saranno forniti nel capitolo seguente.

1.2.2.1 Esempi di applicazioni sviluppate presso il laboratorio PERCRO

Numerose sono le applicazioni che sono state sviluppate utilizzando gli strumenti descritti sopra presso il Laboratorio Percro. In questo paragrafo verranno descritte brevemente alcune, rimandando all'articolo di Carrozzino et al.(2009) per ulteriori esempi o dettagli aggiuntivi.

Un primo esempio è rappresentato dall'Inferno di Dante (Figura 11), un Paesaggio di Informazione costruito collocando ogni canto del poema sulla superficie di cilindri concentrici di diametro decrescente via via che si scende verso il basso, proponendo dunque una struttura spaziale che riflette quella dell'Inferno così come è stato concepito da Dante. La navigazione può essere libera o seguendo dei percorsi definiti attraverso i link.

Un secondo esempio è un Information Landscape denominato Cubispace applicato per l'organizzazione di una collezione di arti grafiche lucchesi fisicamente non esposte al pubblico e distribuite tra differenti istituzioni culturali (Figura 12). Per quanto riguarda l'organizzazione del materiale, la strategia è stata quella di organizzare le varie aree tematiche (acquerello, incisione, disegno, litografia) in cubi disposti nello spazio, e collocare i diversi tipi di informazione su facce diverse del cubo (cenni storici, opere, confronto di opere, materiali e strumenti). Eventuali dettagli sono posizionati dietro la corrispondente faccia utilizzando un Corridor Layout. L'utente è libero di esplorare l'ambiente, scegliere uno dei percorsi predefiniti, esaminare uno specifico contenuto. Per agevolare la navigazione è posto al centro dell'ambiente un menu legenda, con link che portano direttamente alle tematiche desiderate.

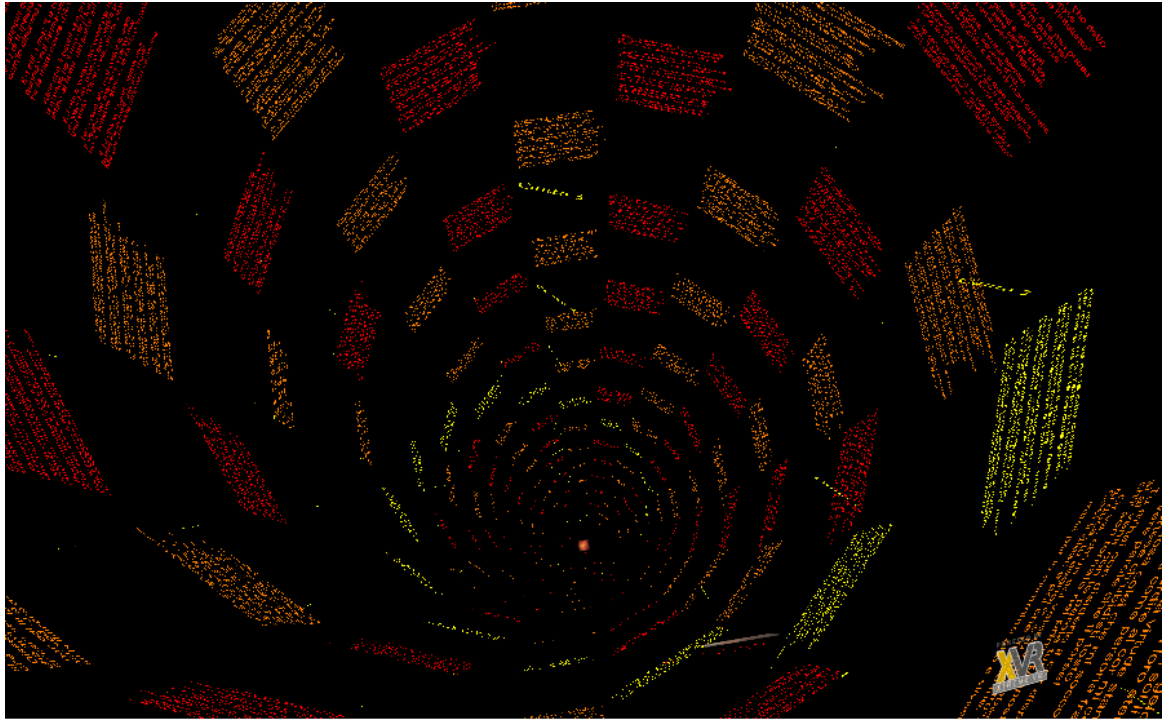


Figura 11 L'IL "DANTE's INFERNO", Lab. PERCRO

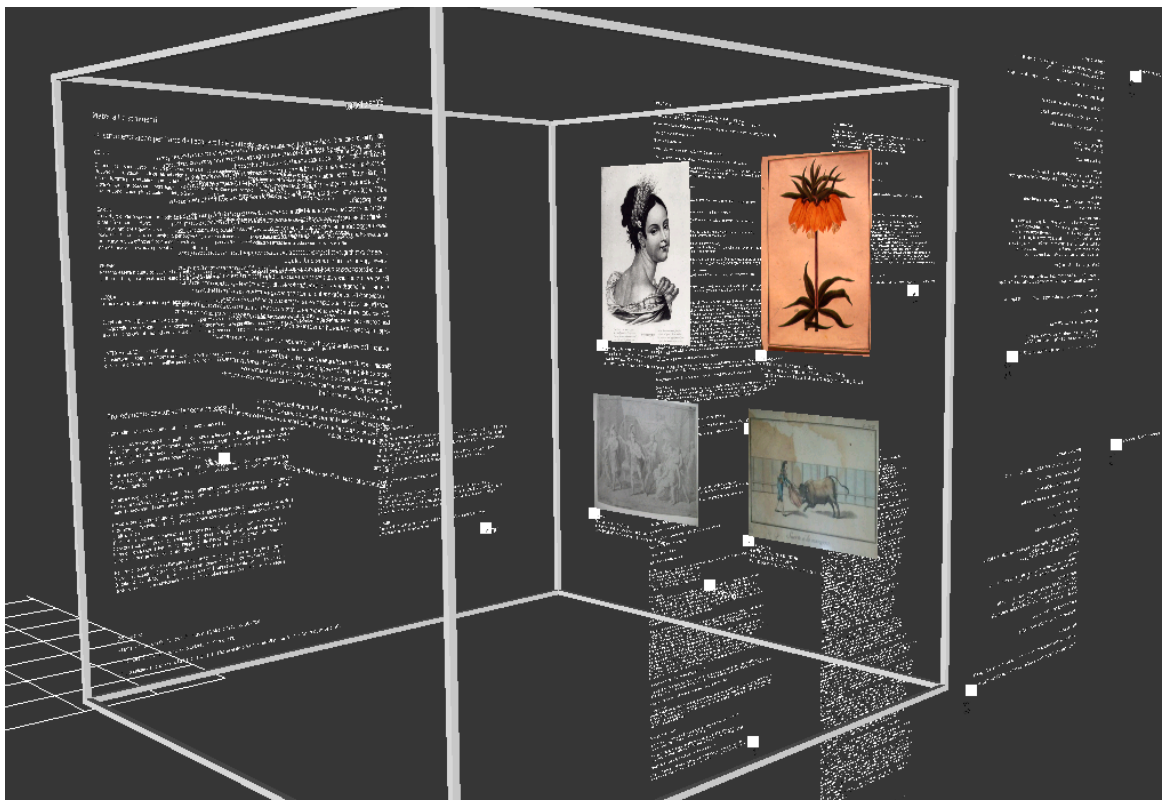


Figura 12 L'IL "Cubispac", Lab. PERCRO

1.2.3 Xanadu Space (Ted Nelson)

XANADU SPACE è l'ultima creazione nata nell'ambito di XANADU, progetto iniziato nella seconda metà degli anni sessanta sotto la guida di Ted Nelson, il riconosciuto padre dell'ipertesto elettronico nonché inventore della parola stessa (Ridi, 1994). XANADU mirava alla realizzazione di un sistema innovativo di organizzazione dei documenti, una rete globale per l'interscambio e la condivisione di testi ed informazioni, una sorta di World Wide Web ante litteram ma più complesso, dato che doveva consentire di tenere traccia delle relazioni tra diverse versioni dello stesso documento, tra i testi derivati e i testi originati, tra le fonti e le citazioni ed inoltre una gestione del diritto d'autore. Dati gli obiettivi così ambiziosi troppo teorici e visionari, fino al 1972 non venne realizzato alcun prototipo e, con l'emergere del web, il progetto cadde lentamente nell'oblio. Tuttavia il progetto non ha mai abbandonato i suoi obiettivi e nel 2007 ha rilasciato un visualizzatore tridimensionale di ipertesti denominato appunto XANADU SPACE (Nelson et al., 2007) e che viene qui riportato perché, nonostante qualche differenza, ricorda in molti aspetti i Paesaggi di Informazione.

Secondo Nelson XANADU SPACE rappresenta un passo verso il modo in cui gli ipertesti dovrebbero essere. Lo scopo dell'ipertesto secondo Nelson è il superamento delle limitazioni della carta, la quale non mostra facilmente le connessioni, ha spazi limitati, e obbliga ad una rigida disposizione rettangolare dei contenuti. Tuttavia, secondo Nelson, fino ad oggi, nell'ambito degli ipertesti e del web nello specifico, hanno dominato vari meccanismi che non hanno portato a sfruttarne le vere potenzialità (imitazione della carta, link monodirezionali e non tipizzati, eccessiva gerarchia, mappaggio semplificato dei contenuti in files).

XANADU SPACE visualizza i documenti all'interno di uno spazio tridimensionale (Figura 13). Ogni documento è formato da una o più unità distinte chiamate vunits e può essere visualizzato in varie forme: pagine rettangolari, cilindriche, in movimento ecc (Figura 14, sinistra). Tra le varie pagine possono essere stabiliti delle connessioni che possono essere di due tipi: collegamenti (links) e transclusioni (transclusions). I collegamenti, le linee chiare presenti in Figura 13, rappresentano appunto una connessione di qualche tipo tra contenuti testuali differenti. Le transclusioni, in rosso, collegano invece tra loro contenuti testuali identici presenti in più documenti differenti. Lo spazio può essere navigato liberamente utilizzando combinazioni di tasti della tastiera e il mouse.

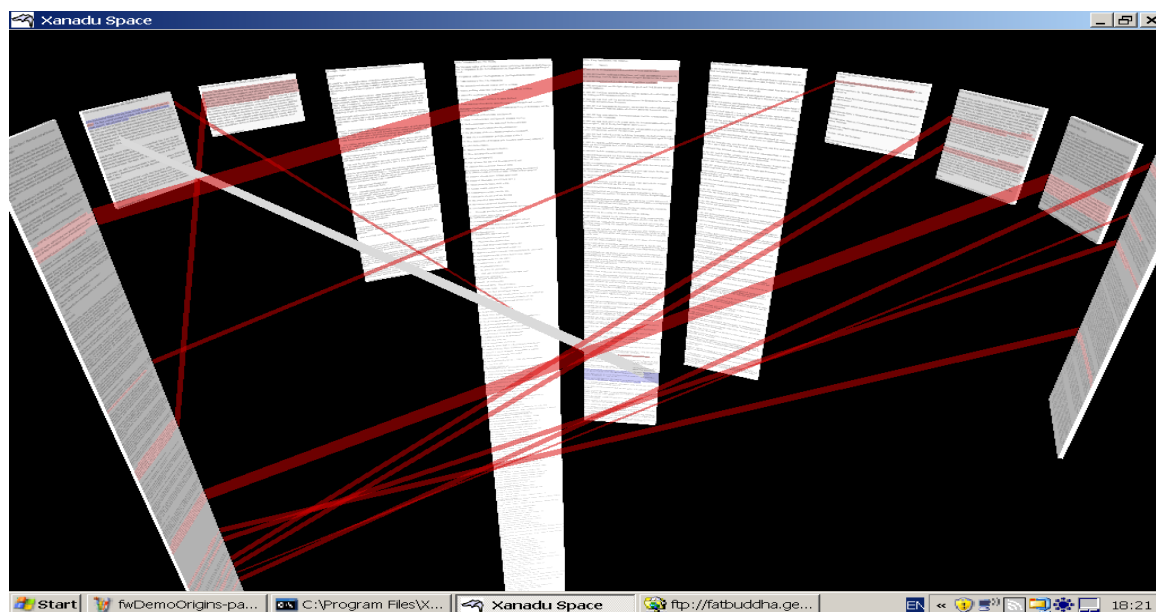


Figura 13 XANADU SPACE (Ted Nelson)

Per facilitare la lettura di una pagina, il raggiungimento di altri contenuti collegati, il confronto con un altro contenuto, e nello stesso tempo mantenere la visione dell'intera scena e di altri eventuali collegamenti, è stato pensato anche un meccanismo innovativo di visualizzazione che è stato chiamato piano di lettura o linea di fuoco (Figura 14, destra). Tale soluzione mostra la pagina corrente al centro dello schermo, insieme ai collegamenti ad altri documenti. Selezionando un collegamento, il contenuto che si trova all'altro capo del link avanza posizionandosi accanto al documento corrente. Questo meccanismo di visualizzazione è utile soprattutto quando siamo in presenza di un elevato numero di collegamenti, che renderebbe difficoltoso seguire un link e porterebbe facilmente nella condizione di essere "lost in hyperspace".

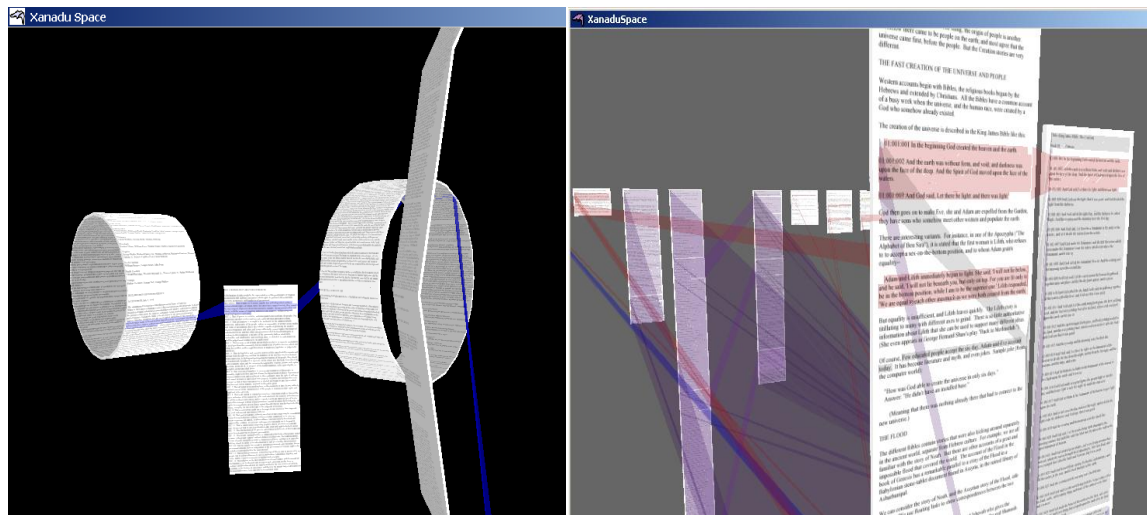


Figura 14 Documenti di varie forme in Xanadu Space (a sinistra), tecnica di visualizzazione detta "piano di lettura" (a destra)

Un video dimostrativo, in cui lo stesso Ted Nelson illustra le potenzialità del sistema, è disponibile sul web².

1.2.4 Un IL per la valorizzazione e divulgazione della storia dell'arte

Doron Goldfarb et al (2011a, 2011b) hanno progettato e implementato un prototipo di applicazione che, a partire da una combinazione di varie sorgenti di informazioni presenti online relative alla storia dell'arte, costruisce automaticamente una visualizzazione tridimensionale applicando la metafora degli Information Landscape. In particolare i dati sono prelevati dalla Web Gallery of Art (WGA)³ che fornisce un'ampia collezione digitale di opere d'arte e alcuni rilevanti metadati, dalla Getty Foundation che fornisce ricchi vocabolari di termini come il Getty Union List of Artist Names (ULAN)⁴, e da basi di conoscenza provenienti direttamente dalla comunità di utenti. L'ambiente è costruito dinamicamente a partire da una query testuale inserita dall'utente. La visualizzazione (Figura 15) mostra gli artisti e le loro opere come nodi e le loro relazioni (es. insegnante/allievo, genitore/figlio ecc.) come collegamenti, il tutto disegnato secondo un ordine cronologico che consente all'utente di esplorare la storia dell'arte in un modo innovativo.

² http://www.youtube.com/watch?v=En_2T7KH6RA7

³ <http://www.wga.hu>

⁴ <http://www.getty.edu/research/tools/vocabularies/index.html>

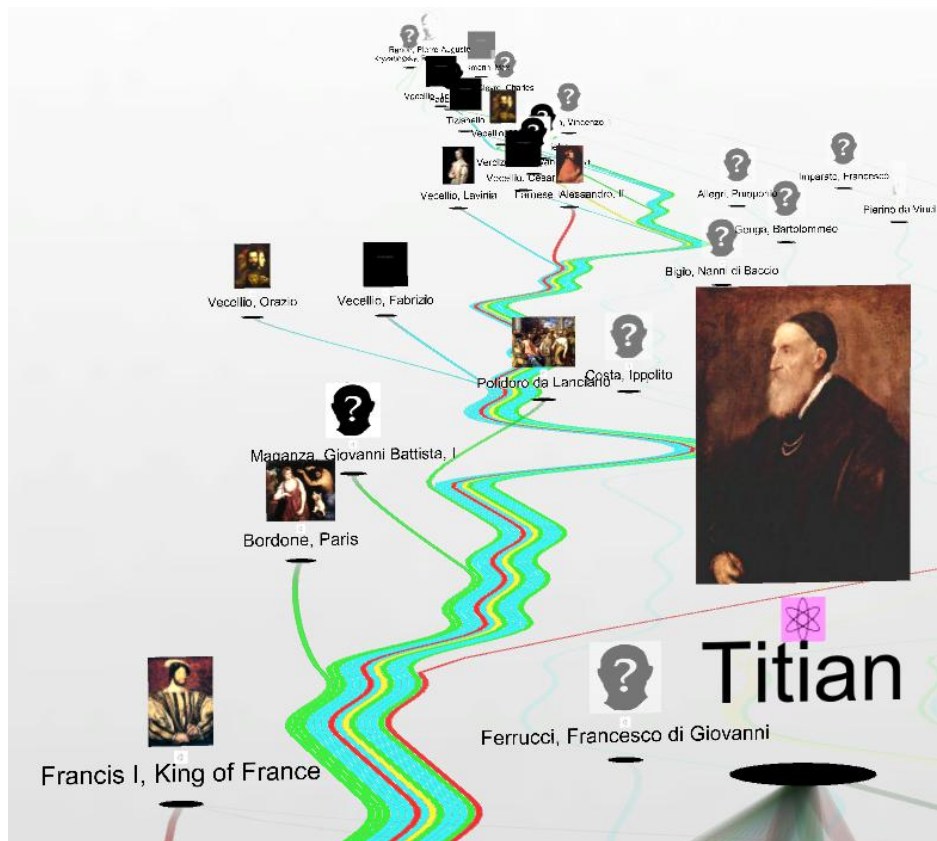


Figura 15 L'IL sulla storia dell'arte di Doron Goldfarb et al (2011a, 2011b)

Nonostante l'astrazione che caratterizza l'ambiente gli autori hanno preferito mantenere qualche riferimento alla realtà (l'uso di una superficie -il terreno- su cui sono poggiate le informazioni, la presenza del cielo sopra) al fine di semplificare la navigazione e l'immersione.

Sono inoltre forniti aiuti alla navigazione come un'etichetta in basso a sinistra che mostra la data storica associata alla posizione corrente dell'utente, una mappa sul lato destro dello schermo che mostra una visione dall'alto dell'IL rendendo più semplice per l'utente localizzare sé stesso nella linea del tempo, una bussola ad ago mostrata in alto a sinistra fornisce un feedback all'utente sulla sua orientazione corrente rispetto al flusso del tempo. Due sono le modalità di navigazione: la prima consente all'utente di muoversi liberamente nell'ambiente, utilizzando il mouse; la seconda consiste nella selezione di un collegamento tra due nodi, operazione che sposta automaticamente l'utente lungo il percorso disegnato dal collegamento in una sorta di animazione.

Dal punto di vista implementativo si utilizza la soluzione proposta in (Wielemaker et al, 2008) per l'integrazione dei dati provenienti dai vocabolari forniti da Getty con le risorse

visuali descritte attraverso i metadati. Inoltre per la memorizzazione dei dati si utilizza la piattaforma ClioPatria⁵, che fornisce varie APIs per il recupero e la presentazione dei dati. Per la visualizzazione si utilizza il game engine Unity⁶, che fornisce un potente ed economico ambiente di sviluppo adatto alla programmazione di ambienti 3D interattivi. La disponibilità di un plugin che può essere installato sul browser permette all'utente di accedere all'ambiente direttamente dal Web.

1.2.5 WWW3D: un esempio di web browser 3D

Come rappresentante di numerosi sistemi che hanno tentato una visualizzazione del Web mediante paesaggio tridimensionale proponiamo WWW3D, un sistema realizzato da Snowdon, Fahlén e Stenius (1996). Le varie pagine sono visualizzate come sfere collocate nello spazio 3D ed etichettate con il titolo della pagina web (Figura 16,a sinistra), mentre il contenuto si trova all'interno della sfera. Selezionando una sfera, l'utente si ritrova al suo interno (Figura 16, a destra); la sfera viene renderizzata in wireframe in modo da poter mantenere una visione dello spazio esterno. L'utente può quindi esplorare il contenuto della pagina o selezionare un link ad altre pagine: tale selezione porterà la nuova sfera che rappresenta la destinazione del link a collocarsi vicina a quella contenente il link. Per tenere traccia della storia dei link visitati, tra i documenti collegati da link è disegnata una freccia la cui brillantezza è proporzionale al tempo trascorso dall'ultima volta che il collegamento è stato attivato. Anche il colore delle sfere fornisce un'indicazione del tempo trascorso dall'ultima volta che sono stati visitati.

Il sistema, realizzato presso lo Swedish Institute of Computer Science (SICS), Svezia, e il dipartimento di Computer Science dell'Università di Nottingham, Gran Bretagna , appoggiandosi al sistema DIVE (Distributed Interactive Virtual Environment) permette a più utenti di collegarsi contemporaneamente, navigare e interagire con gli oggetti presenti, comunicare tra loro.

⁵ <http://e-culture.multimedien.nl/software/ClioPatria.shtml>

⁶ <http://unity3d.com/>

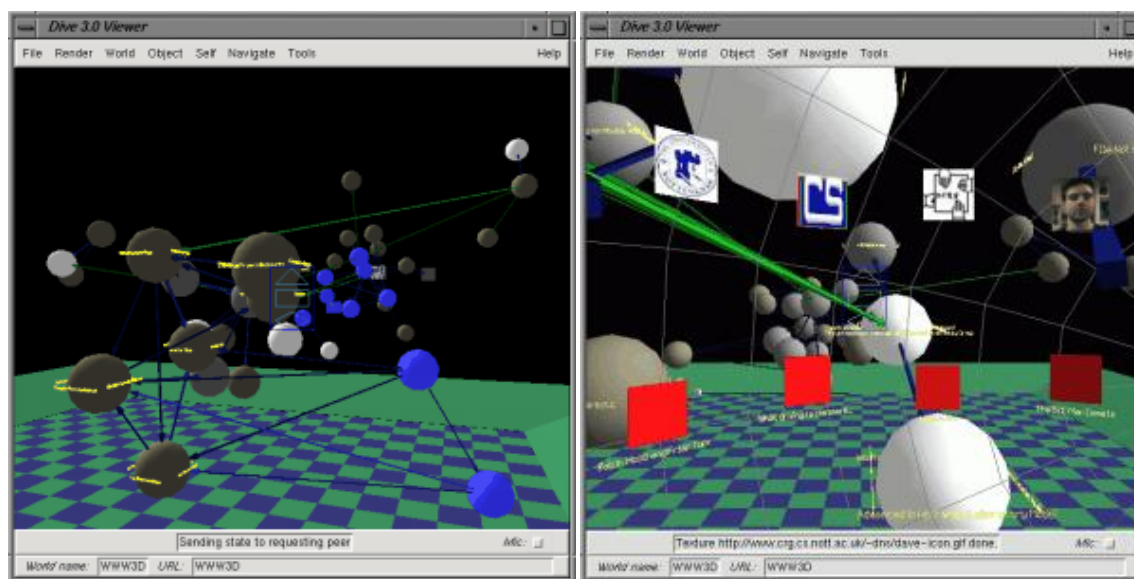


Figura 16 Rappresentazione delle pagine come sfere in WWW3D (a sinistra), rappresentazione del contenuto delle pagine all'interno delle sfere (a destra)

1.2.6 City of News

Sempre nell'ambito della visualizzazione tridimensionale del web proponiamo, per la sua particolarità e per un maggiore interesse rivolto verso la ricerca di nuove modalità di interazione, il progetto City of News.

City of News (Sparacino, 2000) è un browser web 3D sviluppato nel 1997 presso il MIT in cui l'informazione è mostrata come un paesaggio urbano dinamicamente creato e in continua evoluzione (Figura 17). Partendo da una "home page" a scelta, il browser preleva gli URL presenti e costruisce grattacieli ed edifici sulle cui superfici sono collocate le informazioni, generando una città attraverso cui l'utente può spostarsi. La città è organizzata in quartieri urbani (distretti) che forniscono raggruppamenti delle informazioni. Analogamente alle maggiori città contemporanee è presente un distretto finanziario, un distretto dedicato all'intrattenimento e uno dedicato allo shopping, allo sport, alla scienza, ai libri ecc.. Possiamo immaginare questi distretti come le directory disponibili su alcuni dei motori di ricerca più importanti. La città evolve e cresce attraverso l'esplorazione: seguendo un link, viene costruito un nuovo edificio nel distretto di appartenenza in base al contenuto. I nuovi edifici della città sono aggiunti in base ad un algoritmo che simula l'evoluzione e la crescita artificiale.

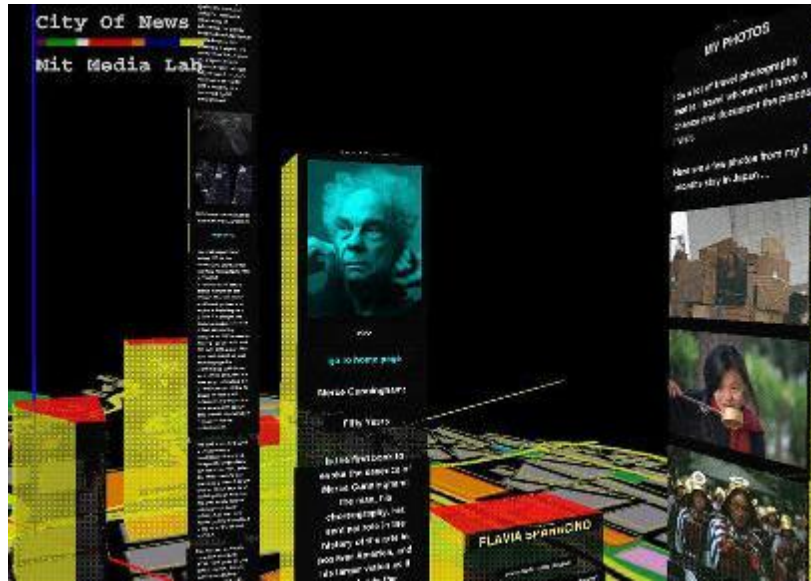


Figura 17 CITY of NEWS, un esempio di paesaggio

L'applicazione può essere sperimentata sia come un browser web di tipo desktop sia in una versione più immersiva denominata Body Driven City of News (Sparacino, 2001) dimostrata al SIGGRAPH 99, che utilizza un'interfaccia di visualizzazione stereografica e traccia i gesti del corpo per interagire. In questo caso, il sistema è costituito da uno schermo stereografico ad alta risoluzione posto dietro ad una scrivania, un sistema audio stereofonico, a cui si affiancano un paio di telecamere che servono per ottenere una rappresentazione 3D della testa e delle mani del visitatore (Figura 18, sinistra). Esiste poi una modalità di fruizione simile alla precedente ma che prevede l'utilizzo di uno schermo molto più grande e la possibilità di stare in piedi di fronte ad esso; sul pavimento è proiettata una mappa della città: camminando sulla mappa è possibile spostarsi nella città (Figura 18, destra).

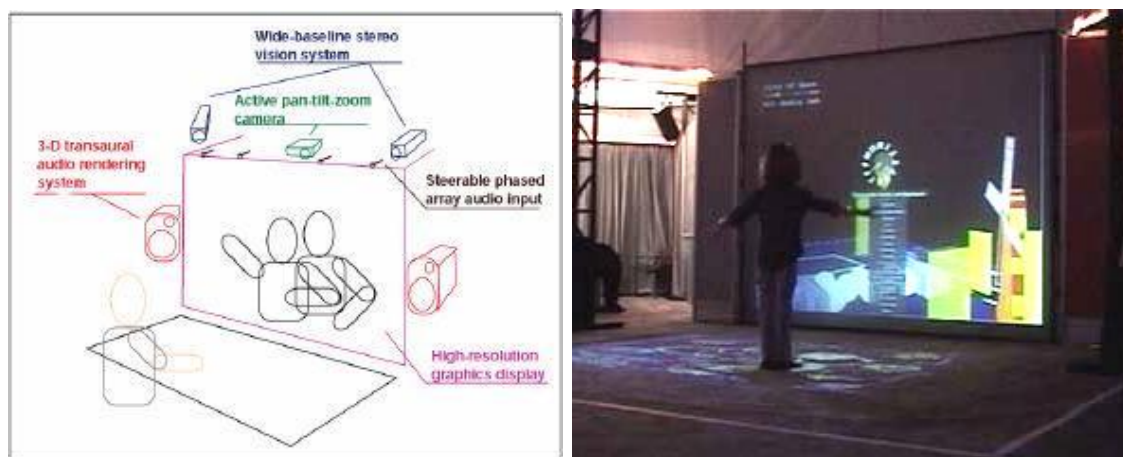


Figura 18 Body driven City of News nelle due versioni: desktop (a sinistra), grande schermo (a destra)

1.3 Tentativo di Classificazione

Dagli esempi analizzati sopra, è possibile tentare una classificazione dei sistemi di Information Landscape in base a diverse variabili. In base alle possibilità di utilizzo possiamo identificare sistemi ad uso generico e sistemi ad uso specifico. Per sistemi ad uso generico intendiamo sistemi che non sono stati pensati e implementati per la presentazione di un contenuto specifico o una tipologia di dati ben definita ma che possono essere usati per visualizzare contenuti di volta in volta differenti. Questi sistemi spesso offrono strumenti di editing per la creazione dei contenuti che poi possono essere caricati e visualizzati dal software di visualizzazione. Rientra in questa tipologia, per esempio, il sistema progetto presso il laboratorio PERCRO, dove si mantiene una forte distinzione tra software per la visualizzazione e contenuti da visualizzare. Per sistemi ad uso specifico, viceversa, intendiamo sistemi che sono stati progettati per la presentazione di un contenuto specifico o di un certo tipo di dati, come l'applicazione che abbiamo visto sopra per la rappresentazione della storia dell'arte.

Dal punto di vista della modalità di creazione della visualizzazione possiamo distinguere tra sistemi automatici e sistemi manuali. Nel primo caso a partire da un insieme di dati memorizzati secondo certi formati, utilizzando algoritmi di analisi di dati e altri per la creazione della rappresentazione grafica, si ottiene automaticamente il paesaggio di Informazione. Il vantaggio di questi sistemi è la velocità di creazione, la dinamicità ovvero il fatto che il paesaggio può cambiare in base alle query dell'utente, la possibilità di portare alla luce nuove relazioni tra i dati. Lo svantaggio è l'assenza di controllo da parte di un designer che in certi casi potrebbe essere in grado di individuare meglio la corretta disposizione spaziale in base al messaggio da trasmettere. Rientrano in questa categoria, per esempio, City of News, WWW3d, l'applicazione per la rappresentazione della storia dell'arte. I sistemi manuali, invece offrono al designer il massimo controllo sulla progettazione del layout informativo e delle modalità di navigazione, e permettono l'inserimento di quel tocco di creatività non possibile diversamente. Il più significativo rappresentante di questa modalità è il sistema progettato al PERCRO che ha permesso per esempio la creazione di ILs come la Divina Commedia in cui l'informazione è disposta secondo la rappresentazione a cono rovesciato caratteristica dell'Inferno di Dante. Alcune linee guida per il design di un paesaggio di informazione sono fornite in Carrozzino et al. (2009) Lo svantaggio di questa modalità è lo sforzo e il tempo necessario alla progettazione e alla realizzazione del paesaggio.

I sistemi di IL differiscono tra loro anche in base alla tipologia di contenuto che sono in grado di gestire. Inizialmente nati come visualizzatori di soli contenuti testuali come nel caso del Typographic Space, gli ILs si sono evoluti consentendo la rappresentazioni di numerosi tipi di dati astratti come immagini, video, suoni, modelli tridimensionali, tanto da poter parlare di Multimedia Information Landscapes (MIL). Rientrano pienamente in questa modalità per esempio il sistema sviluppato presso il Lab Percro e City of News, che supportano pressoché tutte le tipologie di informazione e in modo parziale le altre applicazioni illustrate sopra.

In riferimento alla modalità di fruizione, esistono sistemi per la fruizione desktop e altri per la fruizione immersiva. I sistemi desktop possono consistere in applicazioni stand-alone o in applicazioni fruibili all'interno di un browser previa installazione di un plugin⁷. Le modalità di interazione solitamente utilizzate per la fruizione desktop sono il mouse e la tastiera, dispositivi che pur non confacendosi pienamente all'utilizzo di applicazioni 3D, hanno il vantaggio di essere presenti pressoché in tutti i dispositivi; anche se non mancano soluzioni alternative e più creative come quella realizzata da Small che ha utilizzato i mattoncini LEGO.

Per quanto riguarda i sistemi immersivi, sono state realizzate varie soluzioni: una prima soluzione è quella di utilizzare un grande schermo stereoscopico ad alta risoluzione, un sistema di rendering audio 3D e un sistema per la rilevazione dei gesti del corpo dell'utente e il loro riconoscimento per l'esecuzione di comandi di navigazione (City of News). Un'altra possibilità è quella di utilizzare dispositivi di visualizzazione più complessi come il CAVE. Si tratta della soluzione adottata dal PERCRO lab dove però almeno per il momento l'unica modalità di interazione è l'utilizzo di un Joystick. Soluzioni più complesse che utilizzano per esempio le interfacce aptiche non sono da escludere, tuttavia al momento non sono state realizzate implementazioni.

Osserviamo inoltre che nonostante la diffusione di dispositivi touchscreen sia desktop che mobili (per esempio i tablet), ancora non sia stata sviluppata una soluzione per questi dispositivi.

⁷ È importante osservare come l'approvazione di WebGL come standard per la creazione di applicazioni di grafica tridimensionale interattiva sul web, ha reso obsoleto l'utilizzo di plugin per la fruizione di tali tipi di contenuto. Non siamo a conoscenza di sistemi di ILs che utilizzano WebGL per l'implementazione, ma si tratta sicuramente di un possibile sviluppo attuabile.

1.4 Attuali ambiti di applicazione

L'analisi delle applicazioni di Information Landscape esistenti ci permette di individuare alcuni dei campi di applicazione dei Paesaggi di Informazione.

Un primo campo di utilizzo è la visualizzazione in modo innovativo del Web, dato che è possibile ottenere una rappresentazione visuale della struttura che lega tra loro le pagine, facilitando dunque la comprensione, la ricerca e l'orientamento dell'utente. È interessante notare come il linguaggio che utilizziamo comunemente per descrivere il WWW fa un costante riferimento alla dimensione spaziale: si parla di *siti*, di *indirizzi*, di *home page*, ci incontriamo in *chat rooms*. Tuttavia il modo in cui esploriamo il web è bidimensionale: in un certo senso, c'è una dissonanza cognitiva tra il modo in cui immaginiamo il web, e il modo in cui si accede ad esso (Sparacino, 2000). È importante notare come gli esempi riportati sopra (WWW3d e City of News) rappresentino soltanto due dei tentativi che sono stati fatti di trasporre il web in forma visuale. Altri tentativi di rappresentazione del web sono descritti in Dodge et al. (2001)

Un altro ambito è l'utilizzo nella ricerca, nello studio, e nella divulgazione e nella didattica dei beni culturali. La rappresentazione visiva delle informazioni - soprattutto quella ottenuta automaticamente - relativa ad un determinato ambito possono consentire di rilevare dei collegamenti nuovi, nuovi percorsi mentali e quindi nuove interpretazioni a partire da dati grezzi, come nel caso della strutturazione di dati relativi alla storia dell'arte. Nell'ambito della divulgazione, sia attraverso il Web che all'interno di istituzioni culturali, ci si è da tempo rivolti verso l'utilizzo di strumenti informatici e tecnologici, per esempio gli ipertesti e gli ipermedia e in certi casi si è utilizzata la realtà virtuale per rappresentare repliche o estensioni di ambienti reali (edifici storici, monumenti, musei ecc) o ambienti che, sebbene verosimili, non hanno dei corrispondenti nella realtà. Nel caso in cui l'informazione che vogliamo veicolare presenti una forte componente di astrazione (informazione testuale o immagini), e nel caso in cui vogliamo rendere fortemente evidenti le relazioni tra le entità, l'uso di ambienti virtuali astratti come gli Information Landscape può rivelarsi una soluzione più efficace.

Nonostante gli esempi precedenti riguardino soprattutto l'ambito umanistico, riteniamo che gli ILs si prestino bene per la strutturazione e visualizzazione di dati appartenenti a qualsiasi ambito disciplinare, andando oltre le classiche rappresentazioni ipertestuali. Anche se non ancora provato empiricamente, riteniamo inoltre che la costruzione di un

Paesaggio di Informazione potrebbe rivelarsi utile anche agli studenti durante lo studio, rivelandosi una sorta di evoluzione 3D delle mappe concettuali; oppure nell'insegnamento sia in presenza che a distanza (e-learning), dato che in molti casi essi sono fruibili sul web. Approfondiamo questi ultimi concetti nel seguente paragrafo, in cui si cercano di individuare i vantaggi dei paesaggi di informazione, fornendone una precisa motivazione.

1.5 Vantaggi della rappresentazione spaziale delle informazioni

Studi di psicologia cognitiva, così come l'esperienza quotidiana di ciascuno di noi, provano come sia relativamente facile per la nostra mente, memorizzare la locazione di centinaia di oggetti presenti intorno a noi, nelle nostre case, nel luogo di lavoro, nella nostra città, e come questa facilità si scontri con la nostra scarsa abilità di memorizzare informazioni come poemi, formule matematiche, i dettagli di una storia che abbiamo letto su un libro o in Internet (Sparacino, 2000). Allo stesso modo, posti di fronte ad una quantità di informazioni molto ampia faticiamo a comprenderne l'organizzazione e le relazioni, mentre siamo relativamente abili a costruire una mappa cognitiva rappresentativa del territorio⁸. Questo forse è dovuto al fatto che, dato il ruolo importante e pervasivo che lo spazio gioca nelle attività umane, le attività di percezione, manipolazione, e movimento nello spazio sono in gran parte attività subcoscienti che richiedono un basso carico cognitivo e offrono invece grandi potenzialità (Kuhn et al., 1996).

Fin dall'antichità si è quindi fatto ricorso a metodi di organizzazione e memorizzazione delle informazioni che sfruttassero la nostra abilità di trattare con lo spazio (Sparacino, 2000). Le tecniche di memorizzazione basate sull'associazione di informazioni in locazioni spaziali risalgono almeno al 500 a.C. Una delle più antiche è quella denominata Palazzo della Memoria. Il palazzo della memoria può essere sia un luogo reale (per esempio un'architettura famosa), sia un luogo di fantasia. Le informazioni da ricordare possono essere associate ad oggetti, immagini, simboli, statue collocati all'interno del palazzo. Un'altro esempio di uso della mappatura spaziale come espediente per la

⁸ Il processo di acquisizione di conoscenza spaziale è stato molto studiato in ambito psicologico. Golledge ritiene che il processo di formazione di una rappresentazione interna dello spazio sia indispensabile per permettere un'interazione con il mondo esterno. Toman ha introdotto il termine "mappa" cognitiva per riferirsi a questa rappresentazione interna. Molti sono stati i dibattiti in psicologia e filosofia per stabilire la natura di queste rappresentazioni interne: c'è chi ritiene che siano mappe mentali, proposizioni verbali che descrivono lo spazio, un insieme di rappresentazioni diverse tra esse connesse, oppure se non ci sia alcuna rappresentazione mentale ma solo associazioni simboliche.

memorizzazione delle informazioni, è il Teatro della Memoria (Figura 19), ideato nel XVI secolo dal filosofo rinascimentale Giulio Camillo Delminio che concepì l'idea di un teatro che doveva permettere di memorizzare tutto lo scibile umano, attraverso schemi di memoria associativa. Dal palco si dipartivano 7 gradini, a ciascuno dei quali era associata una diversa immagine, e ognuno suddiviso nuovamente in 7 parti, tante quanti i pianeti, per un totale di quarantanove caselle, ciascuna contrassegnata da una immagine mnemonica desunta dalla mitologia.

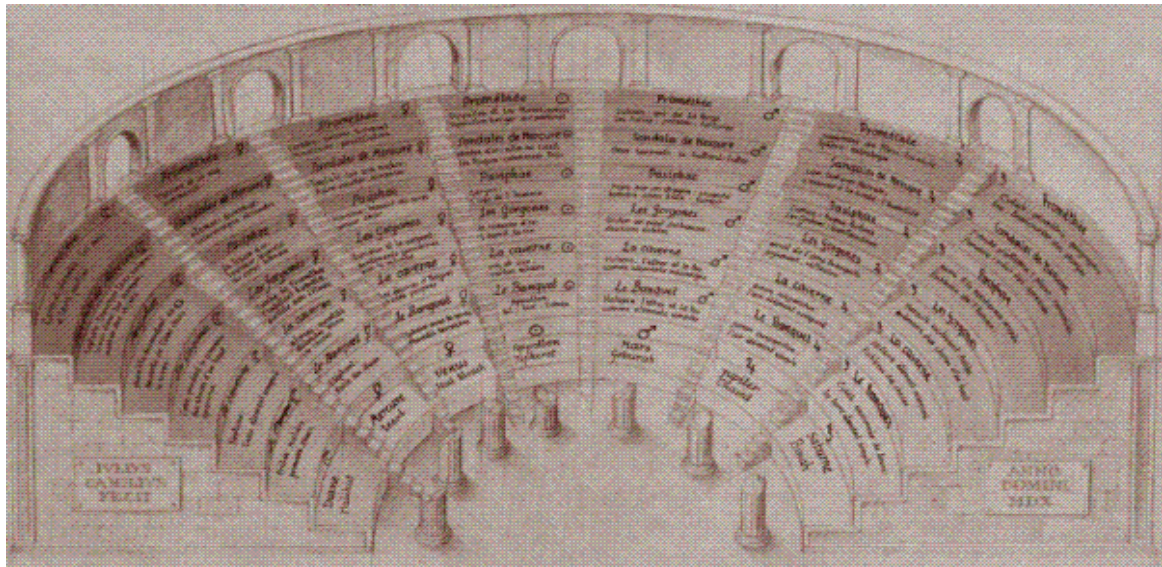


Figura 19 Il teatro della memoria di Giulio Camillo Delminio

Le informazioni, organizzate spazialmente diventano quindi più facili da comprendere, memorizzare, utilizzare come punto di inizio da cui partire per generare nuove idee.

In quest'ottica, anche gli Information Landscapes costituiscono un metodo efficace per organizzare spazialmente le informazioni, mettendone in evidenza possibili strutture e relazioni, con possibili benefici nell'ambito della comunicazione e dell'apprendimento. Possono cioè essere utilizzati come strumento di comunicazione di un determinato concetto, di una determinata struttura informativa. In ambito didattico possono quindi utilizzati nella comunicazione docente-discente. Oppure, in un ottica più costruttivistica, lo studente potrà creare il proprio information landscape come strumento per costruire la propria struttura informativa, partecipando attivamente nella costruzione di conoscenza e favorendo l'apprendimento e la memorizzazione di concetti.

Bisogna inoltre osservare che la struttura proposta dall'information landscape non si impone all'utente in modo estremamente rigido dato che non obbliga quest'ultimo a seguire le relazioni e i percorsi presenti, consentendogli invece di procedere attivamente

con una navigazione libera dell'informazione alla ricerca di una nuova struttura più personale.

Altri vantaggi degli IL emergono se li mettiamo a confronto con i mezzi di comunicazione elettronica precedenti. Inizialmente i media elettronici non sfruttavano tutte le possibilità che la tecnologia poteva fornire: basti pensare alle pagine mostrate in un programma di videoscrittura per comprendere come i media elettronici fossero all'inizio una mera imitazione del supporto cartaceo e del libro tradizionalmente considerato. Un documento word, come un libro è diviso in pagine distribuite sequenzialmente. Un passo in avanti si fa con l'emergere del concetto di ipertesto. Per ipertesto, intendiamo, secondo la definizione di Theodor Nelson (1992, 0/2), "*scrittura non sequenziale - testo che si dirama e consente al lettore di scegliere [...]. Così come è comunemente inteso, un ipertesto è una serie di brani di testo tra cui sono definiti dei legami che consentono al lettore differenti cammini*".

Un problema chiave dell'ipertesto è comunemente chiamato "lost in hyperspace", la perdita dell'orientamento nello spazio. Essere perso nello spazio significa non avere più una conoscenza precisa di dove si viene e dove si può andare, condizione favorita dal fatto che in un ipertesto i link sono invisibili e in un certo istante possiamo visualizzare solamente una determinata pagina perdendone una visione nel contesto (Nelson, 2006). Se è vero che si è cercato di limitare tali problemi attraverso l'invenzioni di aiuti per l'orientamento (come briciole di pane, mappa del sito, navigazione a tab ecc), il problema non è mai stato realmente risolto, o comunque non si è cercato di risolverlo sfruttando tutte le potenzialità che l'informatica mette a disposizione. Inoltre se ci pensiamo bene, l'ipertesto non fornisce una vera e propria "non sequenzialità", ma piuttosto produce una multisequenzialità. L'utente può scegliere ma solo all'interno dei percorsi messi a disposizione con l'autore dell'ipertesto.

I computer, oggi più che mai, possiedono la capacità di trattare con la grafica tridimensionale, permettendo di renderizzare veri e propri paesaggi costituiti da informazioni astratte, come testo e immagini: gli ILs appunto. Questi nuovi linguaggi, con l'uso della grafica tridimensionale potrebbero migliorare la navigazione in quanto permettono di mantenere sempre una visione del contesto. Inoltre, pur proponendo una struttura ipertestuale, di cui rappresentano un'evoluzione, essi permettono anche, senza

bisogno di strumenti aggiuntivi (come i motori di ricerca nel caso del web) , di navigare liberamente tra le informazioni creando dei percorsi personali.

Infine come ultimo vantaggio, bisogna specificare che l'utilizzo di strumenti automatici per la creazione di Paesaggi di Informazione a partire da dati grezzi, può inoltre aiutare a scoprire relazioni tra i dati non identificate in precedenza, permettendo la creazione di nuova conoscenza.

1.6 Impostazione del lavoro di tesi

Sebbene gli Information Landscapes esistano da circa venti anni, e presentino – almeno in teoria – alcuni vantaggi per la modalità con cui presentano l'informazione, nella realtà dei fatti non sono stati comunemente e ampiamente adottati. Tranne alcuni esperimenti svolti in ambito di ricerca, è difficile trovare in giro sistemi di questo tipo.

Varie possono essere le possibili cause: il fatto che non ne sia stata sufficientemente pubblicizzata l'esistenza; il fatto che non siano stati diffusi sistemi che permettano anche ad un utente poco esperto di creare i propri ILs e soprattutto una non accettazione di questa nuova modalità di accesso all'informazione da parte degli utenti.

Se da un lato questa non accettazione potrebbe essere dovuta in parte al fatto che in generale, l'utente medio fatica ad abbandonare il vecchio per il nuovo (per di più passando da due a tre dimensioni), probabilmente la causa maggiore è che finora non si è prestata grande attenzione all'usabilità di tali sistemi, ma si è puntato più sul loro funzionamento dal punto di vista tecnico.

Il nostro lavoro si pone lo scopo di colmare almeno in parte queste lacune a partire dall'attuale soluzione software adottata presso il Lab. PERCRO. In particolare focalizzeremo la nostra attenzione su vari aspetti:

- ampliamento delle funzionalità del visualizzatore Web in modo da fornire anche semplici funzionalità di editor che consentano all'utente la creazione dei propri paesaggi di informazione. Tale sviluppo rappresenta sicuramente un primo passo verso una maggiore diffusione dei paesaggi di informazione verso l'utente comune;
- valutazione dell'usabilità degli attuali sistemi di visualizzazione degli ILs, per mezzo di test con utenti allo scopo di identificare le maggiori problematiche e ipotizzarne delle soluzioni e in alcuni casi implementarle;
- sviluppo di nuove funzionalità che riteniamo possano agevolare l'utilizzo degli IL da parte dell'utente, come per esempio la funzionalità di ricerca full-text.

Siamo convinti che una maggiore attenzione all'usabilità dei sistemi di visualizzazione e delle tecniche di interazione utilizzate con gli ILs, così come lo sviluppo di funzionalità di editor accessibili da web, possano contribuire ad una maggiore accettazione di questa nuova modalità di fruizione dell'informazione e alla sua diffusione, o almeno produrre una conoscenza da un punto di vista innovativo di questi sistemi e delle problematiche ad essi connesse. Di queste tematiche tratteranno i prossimi capitoli della tesi.

2 **Analisi del visualizzatore XVR di Information Landscape (Lab. PERCRO)**

Nel capitolo precedente abbiamo cercato di inquadrare l'argomento degli Information Landscapes, dandone una collocazione nell'ambito delle discipline della realtà virtuale e dell'Information Visualization, presentandone una descrizione accurata e descrivendone lo stato dell'arte.

Abbiamo inoltre evidenziato l'insuccesso in termini di diffusione di questi sistemi, nonostante la loro invenzione risalga ormai a circa venti anni fa, e abbiamo ipotizzato che la creazione di un editor utilizzabile sul web per la creazione degli information landscape, uno studio dell'usabilità, così come l'implementazione di nuove funzionalità, possano portare ad una loro maggiore diffusione e accettazione, o quanto meno possano contribuire a far luce sul perché del loro insuccesso.

Poiché tali sviluppi avverranno prendendo come punto di partenza il visualizzatore XVR sviluppato presso il Lab. PERCRO della Scuola Superiore S. Anna di Pisa, nei paragrafi che seguono, forniremo per prima cosa una descrizione della tecnologia XVR che sta alla base del software, e passeremo poi ad analizzare i dettagli implementativi della specifica applicazione.

2.1 La tecnologia XVR

L'engine per IL è stato sviluppato inizialmente come applicazione stand-alone fruibile in ambiente desktop o immersivo. In anni più recenti è stata però realizzata una nuova versione dell'engine in modo da renderlo eseguibile all'interno di un ambiente XVR.

XVR (Carrozzino, 2005) è un framework tecnologico sviluppato allo scopo di gestire applicazioni interattive e multimediali di realtà virtuale. L'utilizzo di questa nuova

tecnologia consente di fruire degli information landscape non solo come applicazioni desktop o in installazioni immersive, ma anche all'interno di pagine web. XVR infatti consiste in un componente ActiveX in grado di girare sulle piattaforme Windows e che può essere inserito in parecchie applicazioni che fungono da contenitori, incluso il web browser Internet Explorer. Le applicazioni XVR sono sviluppate utilizzando uno specifico linguaggio di scripting (s3d) che fornisce, per mezzo di funzioni e classi specifiche, funzionalità pensate appositamente per la realtà virtuale.

In generale un programma XVR è costituito da 7 funzioni predefinite che vengono eseguite automaticamente. Tali funzioni sono:

- la OnDownload(), eseguita automaticamente all'inizio dell'applicazione, si occupa dello scaricamento all'interno dell'applicazione di tutti i file necessari.
- la OnInit() che contiene il codice di inizializzazione dell'applicazione. Viene chiamata non appena la OnDownload() termina. Riceve come parametro la stringa definita all'interno del codice HTML nella sezione "UserParam".
- la OnFrame(), la sola funzione in grado di produrre output grafico, e di conseguenza l'unica in cui ha senso inserire funzioni e metodi grafici. Essa viene richiamata al termine della OnInit() alla frequenza specificata dalla funzione setFrameRate(), per default stabilita a 100Hz.
- la OnTimer(), richiamata anch'essa al termine della OnInit(), può essere eseguita ad una differente frequenza rispetto alla OnFrame(). La frequenza si stabilisce attraverso la funzione setTimeStep(), e per default è impostata a 100hz. La OnTimer() è il posto adatto per inserire i comandi che devono essere eseguiti in maniera indipendente dal rendering grafico.
- la OnEvent(), viene eseguita ogniqualvolta l'applicazione riceve un messaggio di evento. L'evento può essere esterno (es: i messaggi di Windows) o interno in quanto generato da qualche parte nel programma.
- la DownloadReady(), utile nel caso si vogliano scaricare asincronicamente risorse che non erano necessarie all'inizio del programma. Viene richiamata ogni volta che la funzione FileDownload() è completata e riceve come parametro l'ID del file scaricato.

- la `OnExit()`, richiamata quando l'applicazione termina, è il posto adatto dove gestire la terminazione di eventuali moduli esterni.

Lo script deve essere poi compilato in un bytecode allo scopo di poter essere eseguito dalla XVR Virtual Machine.

Per incorporare un'applicazione XVR all'interno di una pagina Web occorre aggiungere all'interno del codice HTML le seguenti righe di codice:

```
<OBJECT
ID="AppName"
WIDTH = 800
HEIGHT = 800
CLASSID = "CLSID:5D589287-1496-4223-AE64-65FA078B5EAB"
TYPE = "application/x-oleobject" align="left" border="2"
CODEBASE = "http://client.vrmedia.it/XVRPlayer.cab#Version=1,0,0,900">
<PARAM NAME="ScriptName" VALUE=" AppName.s3d.bin">
<PARAM NAME="EngineVersion" VALUE="0136">
<PARAM NAME="BackgroundColor" VALUE="#000066">
<PARAM NAME="EngineParam" VALUE "DEPTH=24">
<PARAM NAME="UserParam" VALUE "...">
</OBJECT>
```

dove il parametro *ScriptName* permette di specificare il nome del file contenente il bytecode dell'applicazione che si intende eseguire; il parametro *engineVersion* permette di specificare la versione di engine XVR che si vuole utilizzare; il parametro *BackgroundColor* permette di specificare il colore di sfondo dell'applicazione XVR; il parametro *EngineParam* permette di specificare alcuni parametri necessari a inizializzare l'engine; il parametro *UserParam* permette di specificare una serie di parametri definiti dall'utente che saranno passati alle funzioni `OnDownload()` e `OnInit()` presenti nell'applicazione.

Quando si accede ad una pagina web che ospita un'applicazione XVR, viene verificata la versione dell'engine richiesta e, se necessario, viene automaticamente scaricata. A questo punto viene scaricato il bytecode dell'applicazione e tutti i dati necessari all'applicazione come modelli 3D, textures, suoni ecc. Eventuali cartelle compresse vengono automaticamente scompattate. Tutte le risorse vengono inserite in una cartella temporanea alla quale è possibile accedere dal programma. Terminata la fase di download delle risorse e di inizializzazione, ha inizio l'esecuzione dell'applicazione vera e propria.

2.2 Il visualizzatore XVR di Information Landscapes

Così come nella versione stand-alone, l'engine XVR per ILs consente di presentare grandi quantità di testo, così come immagini, modelli 3D ecc. Permette inoltre la navigazione nell'ambiente e l'interazione con gli oggetti utilizzando diversi tipi di dispositivi di interazione. Tuttavia l'engine, essendo ancora un prototipo, presenta al momento alcune limitazioni rispetto all'applicazione standalone. In particolare deve essere ancora completata la parte relativa alle animazioni testuali e al supporto di forme di pagine diverse da quella flat tradizionale, così come la gestione dei livelli di dettaglio.

Poiché il software, si presentava privo di commenti e di documentazione che ne agevolassero la comprensione, è stata necessaria una lunga e abbastanza laboriosa attività iniziale di studio del codice, di cui riportiamo sotto i risultati. Riportiamo le parti più significative del codice sorgente nel supporto digitale allegato a questa tesi.

2.2.1 La cartella di progetto: organizzazione del codice

Il codice del visualizzatore è contenuto in una cartella e consiste in una pagina html (*ilview.html*) e un file principale contenente il codice dell'applicazione (*ilview.s3d*). All'interno della sottocartella *include* sono inseriti tutti i file contenente le classi e le funzioni definite per il progetto. In generale, ogni classe è memorizzata su un file diverso in modo da rendere più facile la lettura del codice.

2.2.2 Il diagramma delle classi

Come abbiamo visto, i dati che costituiscono un Paesaggio di Informazione sono logicamente organizzati secondo una struttura ad albero (Cap1, par 1.2.2). Per questo motivo il software costruisce internamente uno scene-graph specificamente pensato per la visualizzazione dei paesaggi di Informazione. Perché questo sia possibile, all'interno del programma sono state definite delle classi ciascuna delle quali definisce la struttura e il comportamento di un'entità che può esistere in un paesaggio di Informazione.

Riportiamo di seguito il diagramma delle classi (Figura 20), e una breve descrizione ad alto livello di ciascuna di essa.

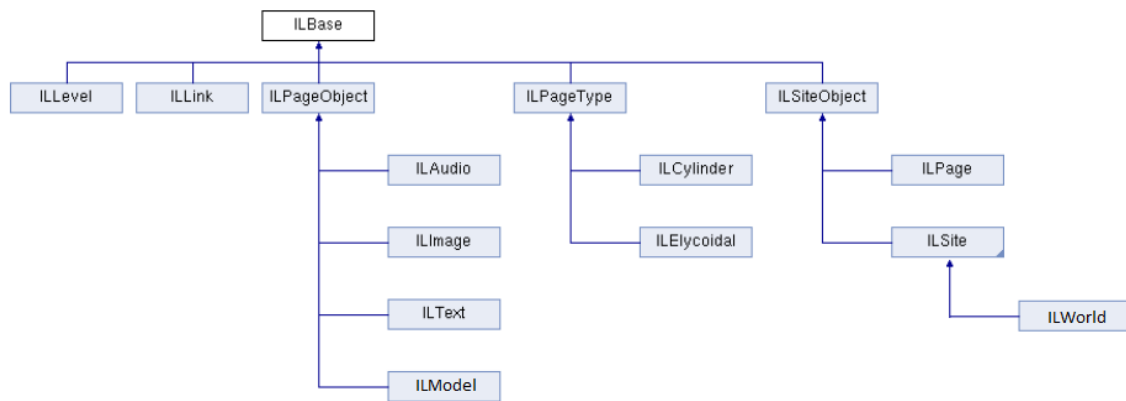


Figura 20 Diagramma delle classi

Le classi che compongono il software sono:

- *ILBase*: fornisce funzionalità di base come la gestione del posizionamento degli elementi, della gerarchia (padre/figlio), dell'identificazione di un elemento attraverso un percorso, del calcolo dei bounding box. Tutte le altre classi derivano direttamente o indirettamente da *ILBase*.
- *ILSite*: definisce le caratteristiche e le funzionalità di un'entità sito (raggruppamento di pagine e/o siti)
- *ILPage*: definisce le caratteristiche e le funzionalità di un'entità pagina (raggruppamento di oggetti informativi)
- *ILSiteObject*: raggruppa le caratteristiche e funzionalità comuni alle entità sito e pagina
- *ILWorld*: derivata da *ILSite*, definisce l'entità di più alto livello ovvero la radice di un information landscape. Essa si può considerare come il Top Level Application Manager e per questo contiene metodi di gestione generale della scena. Se confrontata con il DOM dell'HTML, questa può essere considerata come una combinazione di document e window.
- *ILPageObject*: definisce le proprietà e i metodi generali relativi alle entità (testi, immagini, audio, modelli 3D) che possono essere contenuti in una pagina.
- *ILAudio*, *ILImage*, *ILText*, *ILModel* : derivate da *ILPageObject*, le classi aggiungono le proprietà e i metodi specifici della relativa entità informativa.

- *ILPageType*: contiene le proprietà e i metodi che occorrono a definire un determinato tipo di pagina. Come sappiamo una pagina può assumere varie forme: può essere piatta, cilindrica, elicoidale, ecc. Le sottoclassi che estendono *ILPageType* si riferiscono ad un determinato tipo di pagina⁹.

- *ILLink*: contiene le proprietà e i metodi che servono per gestire la creazione, il rendering e la gestione dei link.

- *ILLevel*: rappresenta un'entità "livello di dettaglio".

Adesso che abbiamo compreso la struttura a classi del software, è importante capire come queste vengono utilizzate e come si combinano tra loro al fine di creare lo scenegraph di un paesaggio di informazione. Più in alto di tutti troviamo un oggetto della classe *ILWorld* che rappresenta la radice e il gestore generale dell'applicazione. L'oggetto *ILWorld* può avere come figli (espressi tramite proprietà *children*) oggetti della classe *ILSiteObject* che come sappiamo possono essere oggetti *ILPage* oppure *ILSite*. Un oggetto *ILSite*, a sua volta, può avere come figli oggetti della classe *ILPage* o altri oggetti della classe *ILSite* (nel caso si voglia esprimere un sito formato da più siti). Un oggetto *ILPage* può avere come figli, oggetti della classe *ILLevel* che servono a esprimere i livelli di dettaglio. Infine, un oggetto *ILLevel* avrà come figli oggetti della classe *ILPageObject* ovvero oggetti di tipo *ILAudio*, *ILImage*, *ILText*, *ILModel*.

È importante notare come nello scenegraph ogni entità presenti il proprio sistema di riferimento, e le entità di più basso livello siano posizionate rispetto al sistema di riferimento del padre.

Perché sia possibile effettuare alcune operazioni di ottimizzazione per il rendering (es. test di culling), il software mantiene costantemente aggiornate opportune strutture dati che memorizzano i *bbox* delle entità che costituiscono il paesaggio, e i frustum necessari (ogni oggetto *ILSite* e *ILPage* presenta infatti un proprio frustum espresso in coordinate locali). Inoltre, per poter supportare altre operazioni (per esempio quelle relative alla gestione dei link), il software determina anche la posizione globale di ogni elemento.

È interessante capire il meccanismo con cui il software tiene aggiornate queste strutture dati (Figura 21). L'aggiornamento dei *bbox* e della posizione globale è necessaria in fase di

⁹ Questa parte non è ancora stata completamente implementata, per cui attualmente è possibile visualizzare solamente pagine "piatte".

caricamento iniziale del paesaggio o nel caso si verifichi un cambiamento nella posizione di un oggetto. L'aggiornamento dei frustum si deve verificare oltre che nel caso appena esposto, anche nel caso di cambio di posizione di camera.

In particolare l'aggiornamento dei bbox avviene secondo una propagazione ascendente (dal basso all'alto) mentre l'aggiornamento dei frustum e il calcolo delle posizioni in world coordinates avviene in modalità discendente (dall'alto verso il basso).

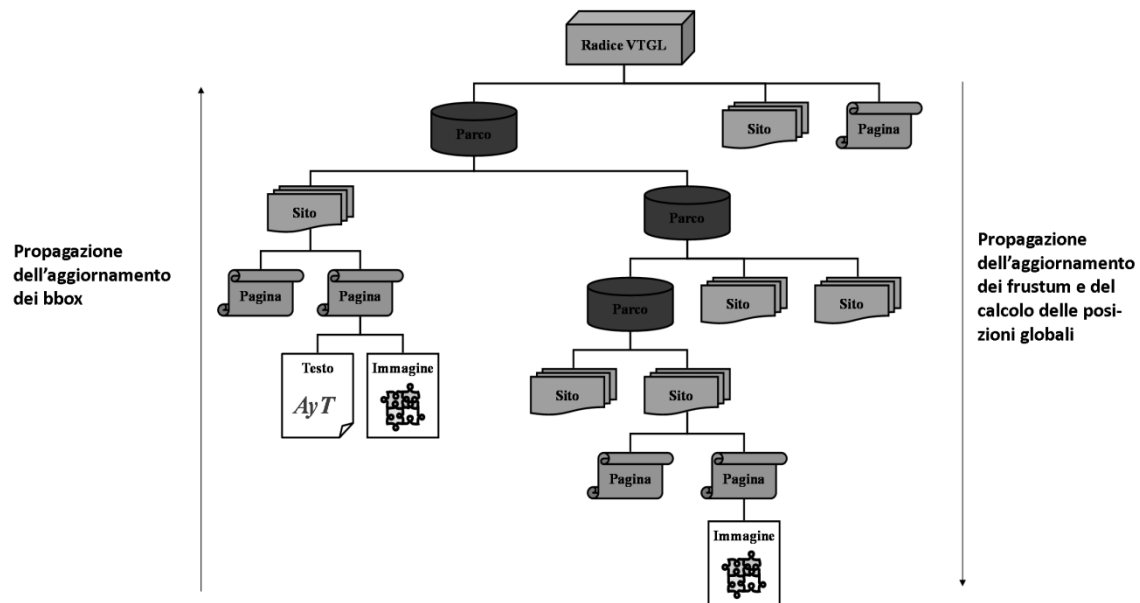


Figura 21 Meccanismo di aggiornamento dei bbox, dei frustum e delle posizioni globali

2.2.3 Gestione del testo

Il testo è renderizzato usando tecniche di texture-mapping, sfruttando per la generazione delle immagini dei glifi la libreria Open Source FreeType. Per maggiori informazioni su tale libreria è disponibile sul web la documentazione relativa¹⁰. In pratica per ogni carattere viene generata una texture la quale viene poi mappata su un poligono rettangolare (quad).

In letteratura esistono anche altri metodi più complessi per la generazione del testo basati su shaders o sulla grafica vettoriale (Qin, 2006 and Loop, 2005) ma a prezzo di maggiori costi e flessibilità del sistema di rendering. La tecnica del texture-mapping è inoltre la più flessibile, adattabile a differenti condizioni e applicabile anche al caso di testo animato o editabile come nella Kinetic Typography (Lee 2002).

¹⁰ <http://www.freetype.org/>

2.2.4 Gestione del rendering

Il rendering viene avviato richiamando il metodo Draw() sull'oggetto ilworld. E' importante sapere che il rendering sarà gestito in maniera differente a seconda di come è settata la variabile globale optListRendering. Nella prima modalità (optListRendering=false), più tradizionale, il sistema fa il rendering percorrendo ricorsivamente lo scene-graph applicando le trasformazioni fino ad arrivare agli oggetti foglia. Nel fare questo, allo scopo di rendere più efficiente il rendering il sistema fa uso di tecniche di culling, le quali consistono nel rimuovere gli oggetti che non rientrano nel frustum di vista¹¹. Nella seconda modalità (optListRendering=true), viene creato preliminarmente un array di tutti gli oggetti che necessitano di essere disegnati (in quanto hanno superato il culling test), e solo dopo lo si scorre tracciando i singoli elementi. Il vantaggio di questa modalità è che, se le trasformazioni nella scena sono poche, esso è molto più veloce in quanto non sono necessarie le chiamate ricorsive che rallentano il processo. Per default viene utilizzata la seconda modalità. È comunque possibile passare all'altra modalità premendo il tasto *l* (lettera L) da tastiera.

2.2.5 Tecnica per il rilevamento della selezione

Per motivi di efficienza, si utilizza la tecnica della selezione per colore anziché la ray-intersection. La selezione per colore prevede che ad ogni oggetto della scena sia assegnato un identificativo unico di colore e che tali oggetti siano memorizzati in una lista. Per stabilire quale oggetto l'utente ha selezionato, è necessario compiere i seguenti passi:

- disabilitare il rendering di texture, lighting e fog
- renderizzare ciascun oggetto usando il suo colore identificativo
- leggere dal framebuffer il colore alla posizione corrente del mouse
- cercare nella lista degli oggetti se c'è un oggetto del colore specificato. Se tale oggetto è presente abbiamo identificato l'elemento selezionato.

¹¹ Al momento il test di culling può essere eseguito solo in modalità di visualizzazione desktop. In modalità immersiva non è presente alcuna ottimizzazione all'interno del software.

La seguente illustrazione (Figura 22) mostra una scena renderizzata normalmente e a destra la stessa scena renderizzata per la selezione per colore.

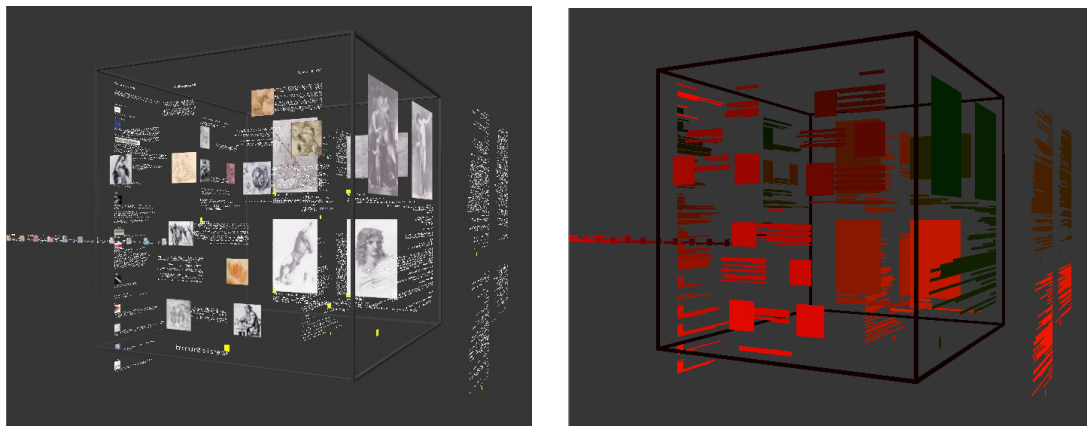


Figura 22 Rendering tradizionale (a sinistra) e rendering di selezione(a destra)

Il meccanismo per il rilevamento di una selezione è gestito all'interno dei metodi *nextSelectionId()* e *processSelection()* della classe *ILWorld*. In particolare, il metodo *nextSelectionId()* si occupa dell'assegnazione di un colore a tutti gli oggetti visibili della scena. Questo avviene memorizzando tali oggetti nell'array *SelectionObject* dove l'indice di posizione corrisponde al colore da utilizzare nel rendering di selezione. La funzione *processSelection()*, invocata dalla funzione *OnFrame()* per tutta la durata dell'applicazione, si occupa di richiamare il rendering di selezione¹², e cioè il disegno degli oggetti con il proprio colore identificativo. A questo punto il metodo *colorSelectionGetIndex()*, preleva il colore alla posizione corrente del mouse e restituisce l'indice di posizione dell'oggetto avente quel colore, se presente. In caso di selezione, verranno invocati opportuni metodi di gestione della selezione.

2.2.6 Contesti di fruizione e dispositivi di interazione supportati

Gli information Landscapes possono essere fruiti sia utilizzando un normale PC oppure, in modalità più immersiva, all'interno del cave. Il CAVE (Cave Automatic Virtual Environment) è un dispositivo di output costituito da tre o più schermi di grandi dimensioni tipicamente retroproiettati che circondano l'utente durante la navigazione dell'ambiente virtuale (Figura 23).

¹² Oltre ai necessari comandi di disabilitazione OpenGL, per richiamare il rendering di selezione è sufficiente impostare la variabile *selectionMode* a *true* prima di invocare il *draw*. In tal modo gli oggetti saranno renderizzati nel framebuffer con il colore di selezione anziché il proprio.



Figura 23 Esempio di fruizione di un IL all'interno di un cave

I dispositivi di interazione supportati sono al momento il mouse, il mouse 3D e il joystick. Solitamente, si utilizza il mouse per la fruizione desktop dato che tutti i computer ne sono dotati, mentre si preferisce utilizzare dispositivi più avanzati come il joystick o il mouse 3D all'interno del cave. Tali devices forniscono infatti un numero maggiore di gradi di libertà rispetto al mouse tradizionale, facilitando in tal modo l'interazione all'interno di un ambiente tradizionale.

La gestione dell'interazione avviene nei file `xmouse.s3d` e nel file `ilinteraction.s3d` e in alcuni metodi della classe `ilWorld` che si occupano di gestire la selezione. In particolare, il file `interaction.s3d` contiene le funzioni per la gestione dei movimenti di camera tramite i vari tipi di dispositivi e una funzione per la gestione degli input da tastiera. Il file `Xmouse` definisce un mouse virtuale, in modo da poter trattare qualsiasi dispositivo (sia esso un mouse tradizionale, un mouse 3d o un joystick) riferendoci ad un solo oggetto con caratteristiche simili al mouse, semplificando quindi la scrittura del codice. La selezione abbiamo visto essere individuata dalla funzione `ILWorld::ProcessSelection()` della classe `ilWorld`. A seconda che venga individuata la selezione di un elemento, nessuna selezione oppure la selezione di un link, verranno richiamate appositi metodi di gestione che sono rispettivamente `ILWorld::processSelectionActive()`, `processSelectionNone()`, `processSelectionLink()`.

Per quanto riguarda l'interazione, vari sono i possibili approcci implementati: navigazione libera nello spazio, navigazione per selezione, navigazione tramite link. La navigazione

libera si basa sulla metafora del volo e permette i movimenti nell'ambiente; la navigazione per selezione consiste nello spostamento automatico del punto di vista in posizione favorevole alla lettura dell'entità selezionata; la navigazione tramite links si verifica selezionando un link, operazione che determina lo spostamento automatico del punto di vista lungo il percorso disegnato dalla curva.

2.2.7 Caricamento di un paesaggio di informazione: il file ilxb

Il visualizzatore non è in grado di caricare direttamente un IL a partire dai file generati dal vecchio editor. Tuttavia è stato sviluppato uno script in linguaggio Python che permette di effettuare la conversione in un formato appropriato con estensione *ilxb*¹³.

Nella figura seguente (Figura 24) si propone una traduzione in formato testuale di un file *ilxb* in modo da comprenderne la struttura.

```
[0,
  'ilx',
  ['DISTANCE',
   [10000.0],
   'REFERENCE_FRAME',
   'AXIS_FULL_GREEN',
   'nodeName',
   'ilx',
   'GRID_COLOR',
   [0.5, 0.5, 0.5],
   'GRID_DIMENSION',
   [20.0],
   'back',
   [0.0, 0.0, 0.0],
   'INIT_ANG_X',
   [0.0],
   'INIT_ANG_Y',
   [0.0],
   'GRID_DIVISION',
   [0.0],
   'startPos',
   [148.0, -1270.0, 438.0]],
 [1,
  'Canto_01.cfg',
  'site',
  ['position',
   [0.0, 120.0, 0.0],
   'DISTANCE',
   [10000.0],
   'nodeName',
   'site',
   'file',
   'Canto_01.cfg',
   'orientation',
   [22.5, 0.0, 1.0, 0.0]],
 [2,]
```

Figura 24 il file *ilxb* (traduzione in formato testuale)

Come si può osservare, ogni elemento che compone il paesaggio di informazione è espresso nella forma [id progressivo, nome, tipo, [proprietà], [figli]] dove proprietà è espressa come sequenza di coppie nome, valore come ["position",[1,2,3],"color",[0.5,0,0],...] e dove figli è costituito da uno o più elementi espressi nel formato specificato sopra e separati da virgola.

¹³ Si tratta di un file binario

L'incorporamento dell'applicazione all'interno della pagina web avviene secondo le istruzioni specificate nel paragrafo 2.1, avendo cura di settare correttamente alcuni parametri specifici. Particolare attenzione deve essere posta all'elemento param avente proprietà *name* UserParam, e la cui proprietà value dovrà contenere una stringa di parametri separati da virgola.

I parametri più importanti sono:

- **local**: indica che vogliamo eseguire l'applicazione in locale
- **path**: utilizzabile solo in localmode, indica la locazione in cui si trovano i dati dell'IL da visualizzare
- **datafile**: utilizzabile solo in modalità non locale, indica il percorso del file zip contenente i dati dell'IL
- **prefix**: prefisso da aggiungere opzionalmente a tutte le risorse caricate
- **ilxb**: indica il nome del file di descrizione del paesaggio di informazione
- **noculling**: disabilita l'operazione di frustum culling test, per esempio nel caso si effettui una visualizzazione all'interno del cave.

Nel caso si voglia eseguire l'applicazione localmente, sarà necessario inserire all'interno della cartella di progetto la cartella contenente il file ilxb che descrive il paesaggio di informazione, insieme a tutte le risorse necessarie (tra cui i moduli esterni e le dll utilizzate dal software). Ammettendo per esempio che tale cartella sia denominata Cubispace e che il file descrittivo sia denominato Cubispace.ilxb la linea di codice HTML riguardante il parametro "UserParam" sarà:

```
<PARAM NAME="UserParam"VALUE="local,path=/Cubispace,prefix=,ilxb=Cubispace.ilxb">
```

Invece, nel caso in cui si voglia caricare la pagina web e l'applicazione in essa contenuta da un server web, sarà necessario inserire all'interno della cartella di progetto una cartella zip contenente il file ilxb e tutte le risorse necessarie. In questo caso non è necessario inserire i moduli e le librerie dinamiche esterne dato che verranno caricate automaticamente dal file ildata.zip già presente nella cartella. Ammettendo per esempio che tale cartella sia denominata DivinaCommedia.zip e che il file di descrizione sia denominato DivinaCommedia.ilxb, la linea di codice HTML riguardante il parametro "UserParam" sarà:

```
<PARAM NAME="UserParam"VALUE="datafile=DivinaCommedia.zip,prefix=,ilxb=DivinaCommedia.ilxb">
```

A questo punto viene avviata l'applicazione. Per prima cosa verrà eseguita la funzione `OnDownload()` all'interno della quale viene effettuato il parsing dei parametri passati e vengono scaricati i file e le risorse necessarie. Segue la funzione `OnInit()` che dopo una serie di inizializzazioni, si occupa di richiamare una funzione di caricamento dei vari elementi che costituiscono il paesaggio di informazione a partire dal contenuto del file `ilxb`. Nel caso non sia stato specificato alcun file, verrà caricato un IL di default che consiste semplicemente nel messaggio testuale "Hello World from ILXVR". Seguono alcune funzioni che è necessario chiamare ogni volta che carichiamo un paesaggio di informazione e che si occupano di aggiornare le strutture dati necessarie al corretto funzionamento del programma (calcolo dei `bbox`, calcolo della posizione globale degli elementi ecc). A questo punto tutto è pronto per l'esecuzione dell'applicazione vera e propria con l'invocazione ripetuta della funzione `OnFrame()`, la quale si occupa di richiamare le funzioni di gestione dei dispositivi di interazione, le funzioni per il rendering tradizionale e di selezione.

3

Sviluppo di un sistema di editing web per la costruzione di Information Landscapes

Nel primo capitolo, abbiamo identificato una delle possibili cause della scarsa diffusione degli ILs presso il grande pubblico nel fatto che, fino ad oggi, non siano stati abbastanza diffusi sistemi che permettano ad un utente di creare i propri Information Landscapes. Abbiamo perciò ipotizzato che l'implementazione di funzionalità di editing accessibili dal web possa rappresentare un primo passo verso una maggiore diffusione dei paesaggi di informazione.

Quando si parla di funzionalità di editing ci si riferisce alla possibilità di creare nuovi elementi, cancellare o modificare quelli esistenti, utilizzare strumenti di manipolazione degli elementi (traslazione, scalatura, rotazione), consentire il salvataggio di un IL su server web e il suo caricamento.

Oltre allo sviluppo delle funzionalità di editing si è cercato di potenziare il visualizzatore attraverso l'aggiunta di uno strumento di ricerca testuale che consente di identificare rapidamente l'informazione testuale di interesse.

Nel paragrafo che segue viene descritto ad alto livello il software realizzato, spiegando come esso possa essere utilizzato per la creazione di un nuovo paesaggio di informazione o per la modifica di uno esistente. Successivamente passeremo ad una descrizione degli strumenti utilizzati e delle scelte progettuali e realizzative effettuate per ogni singola funzionalità.

3.1 Vantaggi di un sistema di editing web

Lo sviluppo di un sistema di editing web ha il vantaggio, rispetto all'editor tradizionale già esistente, di essere più facilmente distribuibile. Infatti, un'applicazione web ha più probabilità di diffondersi rispetto ad un programma che deve essere scaricato e installato sul proprio PC (come ad esempio il vecchio editor).

Inoltre, gli IL creati con questo sistema sono direttamente fruibili da XVR , a differenza degli ILs prodotti con il vecchio editor che necessitavano di un processo di conversione. Questo rende i Paesaggi di Informazione realizzati direttamente fruibili in qualsiasi modalità: desktop, immersiva e web.

In generale, un sistema così pensato inoltre apre nuovi orizzonti verso la creazione e condivisione dei Paesaggi di Informazione con altri utenti.

3.2 Descrizione di alto livello del software realizzato

Il sistema è realizzato utilizzando un'architettura client server. L'applicazione necessita quindi di essere installata su un server web che supporti il linguaggio di scripting PHP¹⁴. Una volta installata, l'applicazione è accessibile utilizzando il browser web Internet Explorer.

L'applicazione consiste in una pagina web contenente una finestra di visualizzazione della scena tridimensionale e un' interfaccia costituita da vari controlli che permettono di compiere varie operazioni di editing e non solo (Figura 25).

¹⁴ Il sistema è stato sviluppato utilizzando la versione 5.4.6 di PHP. Per il funzionamento è' necessario inoltre che sia installata e abilitata la libreria php ZipArchive sul serve.

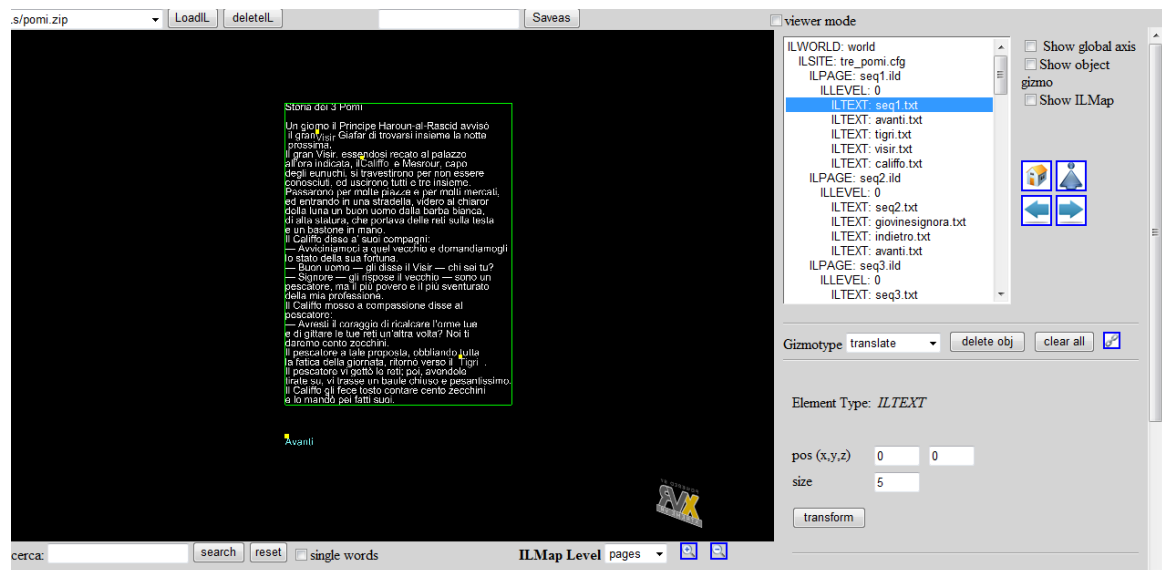


Figura 25 Screenshot dell'editor di IL sviluppato

3.2.1 La 3D view

Al centro della pagina è posta la finestra di visualizzazione della scena tridimensionale (3D view). Essa mostra il paesaggio che si sta realizzando, modificando o semplicemente visualizzando. Il paesaggio può essere navigato liberamente all'interno della 3d view. La finestra può essere inoltre utilizzata per compiere direttamente alcune operazioni sugli oggetti presenti: un doppio clic su un oggetto determinerà la selezione di un elemento; inoltre come vedremo in seguito, sono stati creati dei manipolatori di posizione, scalatura e rotazione degli elementi utilizzabili direttamente all'interno dell'ambiente 3D. Di default, al momento dell'apertura la 3D-view mostra un IL semplice costituito da un solo elemento testuale recante la scritta "Hello World from ILViewer". Esso può essere utilizzato come punto di partenza per la creazione del proprio IL.

3.2.2 Modalità di editing, modalità di viewer

Dato che lo scopo era ottenere un software che fungesse sia da visualizzatore che da editor, si è deciso di separare le due modalità.

In modalità editor (modalità impostata di default all'apertura della pagina web), viene visualizzata un'apposita interfaccia di editing; inoltre, l'interazione e la visualizzazione della vista 3D è diversa rispetto a quella presente nella modalità viewer, in quanto pensata appositamente per agevolare le funzionalità di editing. Per esempio, come vedremo nel

seguito di questo capitolo, la modalità di editing supporta la visualizzazione dei bbox per gli elementi selezionati, la visualizzazione dei gizmo per la manipolazione degli elementi, e altre funzionalità.

Per passare da una modalità all'altra si utilizza la check-box *viewer mode* presente in alto a destra della 3Dview.

3.2.3 Selezione di un elemento

La selezione di un oggetto può avvenire in due modi. Il primo, utilizzabile solo per gli elementi visualizzati, consiste nella selezione dell'oggetto tramite doppio clic sullo stesso all'interno della vista 3D. Il secondo modo, utilizzabile per tutti gli elementi presenti nel paesaggio di informazione (anche quelli non direttamente visibili come siti e pagine), consiste nell'utilizzo della vista di struttura (vedi paragrafo 3.2.4) attraverso un clic sulla opzione desiderata. In entrambi i casi, al fine di rendere chiaro all'utente quale oggetto ha selezionato, sarà visualizzato in verde il bbox relativo (Figura 26) all'interno della vista 3D.

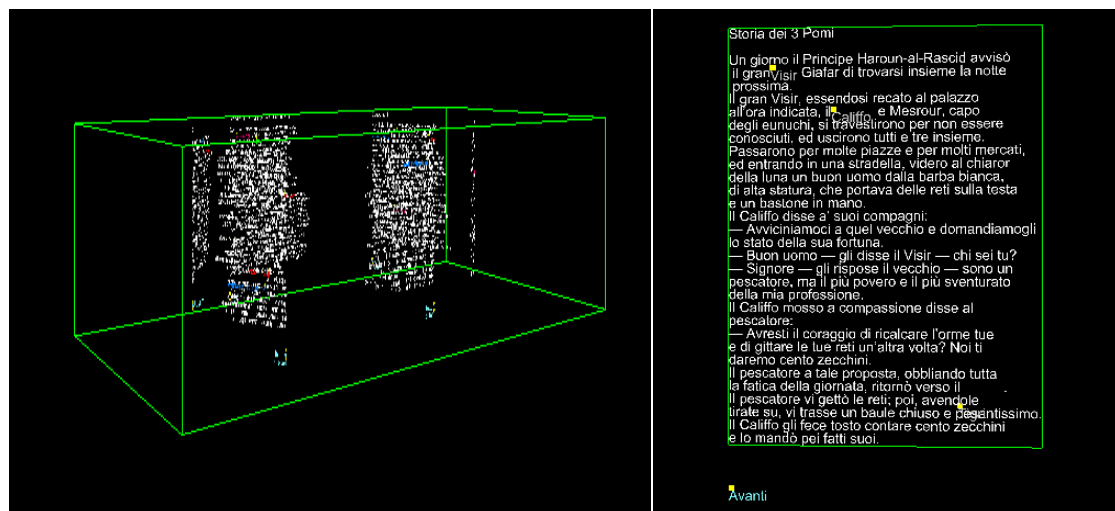


Figura 26. Selezione di un site (a sinistra), selezione di un testo (a destra)

3.2.4 La vista struttura

Al fine di fornire all'utente una migliore comprensione della struttura logica del paesaggio di informazione che sta visualizzando, e di facilitare la strutturazione e la creazione di nuovi paesaggi di informazione si è ritenuto opportuno fornire una *Vista di Struttura* che mettesse in risalto le relazioni gerarchiche che intercorrono tra le varie componenti del

paesaggio (Figura 27). Queste relazioni non possono essere messe in luce in modo chiaro e completo dalla classica visualizzazione 3D. Inoltre tale vista è necessaria per permettere all'utente di selezionare un oggetto su cui eseguire operazioni (settare proprietà, definire elementi figli, ecc) . Tale funzionalità è ancora più importante se pensiamo che all'interno della 3Dview solo gli elementi al più basso livello gerarchico (testi, immagini ,modelli tridimensionali) vengono

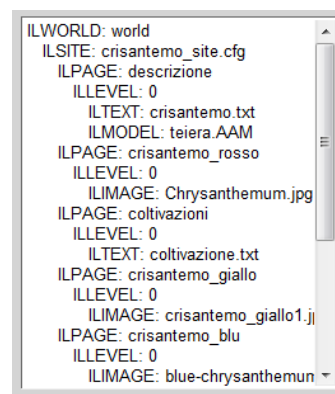


Figura 27. Vista Struttura

visualizzati e quindi possono essere selezionati, mentre per gli elementi ad un più alto livello gerarchico tale modalità di selezione risulterebbe impossibile.

La vista di struttura è resa disponibile tramite una select-box: i vari elementi sono riportati come *options* con il nome che è stato dato loro in fase di creazione (preceduto dal tipo di elemento), mentre la gerarchia è espressa mediante l'utilizzo di un'opportuna indentazione. Gli elementi avranno una maggiore indentazione quanto più appartengono ad un livello gerarchico inferiore; elementi che appartengono allo stesso livello gerarchico presenteranno la stessa indentazione.

La *Vista di Struttura* è disponibile sia in modalità editor che viewer.

3.2.5 Controlli di navigazione

Alla destra della Vista di Struttura, troviamo alcuni bottoni di navigazione (Figura 28). Essi consentono di navigare velocemente alla posizione iniziale, raggiungere automaticamente l'oggetto selezionato nella *Vista di Struttura*, tornare automaticamente sull'ultimo elemento visitato, ritornare avanti nella history degli elementi visitati.



Figura 28 Bottoni di navigazione

I controlli di navigazione, così come la vista di struttura, sono disponibili sia in modalità editor che in modalità viewer, in quanto utili in entrambe le modalità.

3.2.6 Attivatori di visualizzazione assi globali e gizmos

Alla destra della vista di struttura abbiamo anche tre checkbox (Figura 29) che permettono di attivare o disattivare la visualizzazione degli assi globali della scena (Figura 30), il gizmo dell'oggetto selezionato o la visualizzazione della

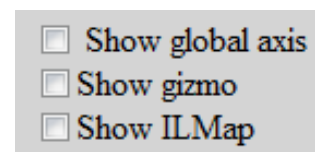


Figura 29 check box per l'attivazione e disattivazione della visualizzazione di assi globali, gizmos, e mappa.

mappa¹⁵.

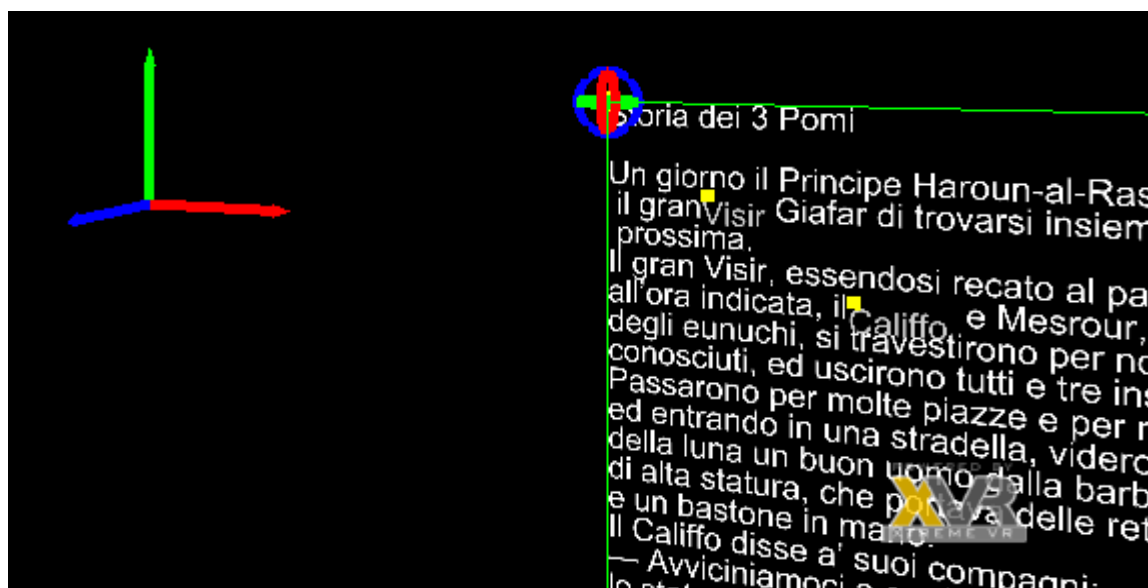


Figura 30 Visualizzazione degli assi globali della scena, e visualizzazione del gizmo di rotazione dell'oggetto testuale selezionato

Gli assi globali sono fissi e servono solo a fornire una visione del sistema di riferimento globale della scena. Nel disegno degli assi sono stati utilizzati i colori solitamente usati nei vari programmi di modellazione: rosso per l'asse x, verde per l'asse y, blu per l'asse z.

I gizmos, a differenza degli assi globali, sono elementi sensibili, nel senso che possono essere attivati con il mouse e utilizzati per ottenere una traslazione, una rotazione o una scalatura di un oggetto. Una funzionalità indispensabile che qualsiasi editor deve mettere a disposizione è infatti quella di consentire la manipolazione degli elementi presenti. Una soluzione divenuta abbastanza comune nell'ambito delle interfacce 3D desktop (per esempio nei programmi di modellazione 3D) è quella di utilizzare dei manipolatori spesso chiamati handles (maniglie) o gizmos renderizzati direttamente all'interno della scena 3D che, se attivati, consentono di esercitare manipolazioni sull'elemento selezionato. Di questi manipolatori esistono svariate versioni con aspetto e funzionamento leggermente diverso. Noi abbiamo preso spunto da quelli messi a disposizione dal software di modellazione 3D Blender.

In modalità editor, selezionando un elemento, se la check box *Show gizmo* è attiva, viene mostrato il gizmo di manipolazione dell'oggetto selezionato. Per default viene

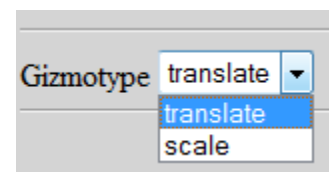


Figura 31 Select box per la scelta della trasformazione desiderata

¹⁵ Della mappa, si parlerà diffusamente nel capitolo 4 della tesi par 4.4.4

mostrato il gizmo di traslazione, ma è possibile attivare un altro manipolatore tramite la select-box *Gizmotype*, nella parte alta del *Pannello di Editing* (Figura 31). È importante notare che non tutte le operazioni (traslazione, scalatura e rotazione) sono disponibili per tutti gli oggetti e che le trasformazioni possibili dipenderanno dal livello in cui l'oggetto si trova nella gerarchia. Per esempio se un oggetto si trova in fondo alla gerarchia (es: *iltext*, *ilimage*) saranno su di esso attivabili solamente la traslazione e la scalatura. La selectbox mostra quindi solamente le trasformazioni disponibili.



Figura 32 Gizmo di traslazione

Il gizmo di traslazione (Figura 32) consiste in tre assi orientati colorati (asse x in rosso, asse y in verde, asse z in blu), ciascuno dei quali se attivato consente di effettuare una traslazione nella direzione dell'asse. In particolare per selezionare un asse è necessario un clic con il tasto sinistro su tale asse, mentre per effettuare lo spostamento è necessario, senza rilasciare il clic del mouse spostare il cursore nella direzione dell'asse. L'elemento seguirà lo spostamento del puntatore del mouse.

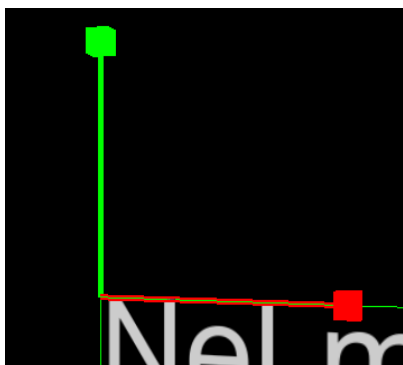


Figura 33 Gizmo di scalatura

Il gizmo di scalatura (Figura 33) è simile nell'aspetto a quello di traslazione con la differenza che agli estremi degli assi anziché essere presenti delle frecce sono presenti dei piccoli cubi. Analogamente, facendo un clic sull'asse ed effettuando uno spostamento del cursore del mouse lungo l'asse senza rilasciare il bottone del mouse otterremo come effetto una scalatura dell'elemento.

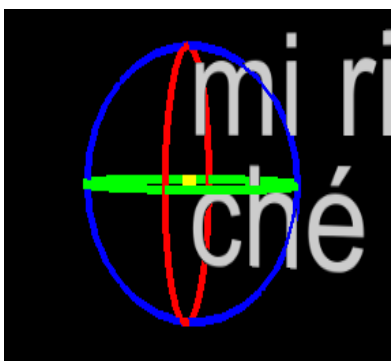


Figura 34 Gizmo di rotazione

Completamente diverso nell'aspetto è il gizmo di rotazione (Figura 34) che consiste in tre cerchi concentrici, uno (in rosso) che rappresenta la direzione di rotazione intorno all'asse x, uno (in verde) che rappresenta la direzione di rotazione intorno all'asse y, e infine uno (in blu) che rappresenta la direzione di rotazione intorno all'asse z. Se si vuole applicare una rotazione ad un oggetto, sarà necessario selezionare

tramite clic la direzione di rotazione, e poi, senza rilasciare il bottone del mouse, spostare il puntatore orizzontalmente.

3.2.7 Pannello di editing

Sempre alla destra della 3Dview troviamo il *Pannello di Editing* dove sono mostrate tutte le possibili operazioni di editing. Il pannello di editing è suddiviso in due parti: la parte alta mostra le operazioni di editing comuni a qualsiasi elemento dell'IL, quella in basso mostra le operazioni possibili per l'elemento correntemente selezionato.

Parte alta del pannello di editing: le operazioni comuni

La parte alta del pannello di editing (Figura 35) oltre alla selectbox per la scelta del tipo di manipolatore di trasformazione da visualizzare, consente la cancellazione dell'oggetto selezionato, la cancellazione di tutta la scena oppure la creazione di un link.

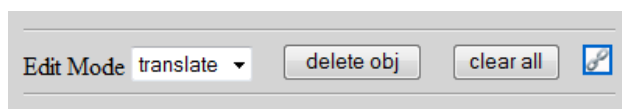



Figura 35 Parte alta del pannello di editing

Il bottone *delete obj* serve per la cancellazione dal paesaggio dell'oggetto selezionato.

Il bottone *clear all* può essere utilizzato per cancellare tutti gli elementi che costituiscono il paesaggio di informazione, con l'effetto di ottenere una scena completamente vuota nella vista 3D. Nella Vista di Struttura sarà mantenuta solo la radice del paesaggio di Informazione denominata *world*. Il comando *clear all* è utile quando vogliamo creare da zero un paesaggio di informazione.

Inoltre, tra due oggetti presenti nella scena è possibile stabilire un collegamento (link). La procedura di creazione di un link prevede la selezione dei due oggetti sorgente e destinazione del link. Il primo oggetto si seleziona tramite doppio click su di esso nella scena 3D; per la selezione del secondo oggetto si effettua doppio click su un altro oggetto tenendo premuto contemporaneamente il tasto SHIFT della tastiera. La prima selezione sarà indicata da un bbox di colore verde, mentre la seconda selezione sarà indicata da un bbox di colore blu. A selezione avvenuta è possibile stabilire un link con un click sull'immagine . Al momento non è possibile creare link multi-ancora (ovvero un link che non ha una sola destinazione, ma si configura come un percorso che attraversa più

elementi nell'ambiente), sebbene il software sia predisposto per tale genere di link. Facendo click su un link apparirà un'interfaccia di editing dello stesso che permette di impostarne il grado della curva, la posizione finale di camera (espressa relativamente alla posizione dell'oggetto destinazione), applicare le nuove impostazioni, visualizzare un preview della posizione che assumerà la camera alla fine del link, oppure effettuare la cancellazione del link corrente.

Parte basso del pannello: le operazioni specifiche dell'oggetto selezionato

Al di sotto del primo blocco di controlli appena descritti, il *Pannello di Editing* mostra tutte le operazioni che possono essere effettuate sul particolare oggetto selezionato.

Selezionando un oggetto mediante una delle modalità di interazione descritte, le sue proprietà vengono visualizzate dal pannello di editing come un insieme di componenti grafici posti in sequenza verticale (Figura 36).

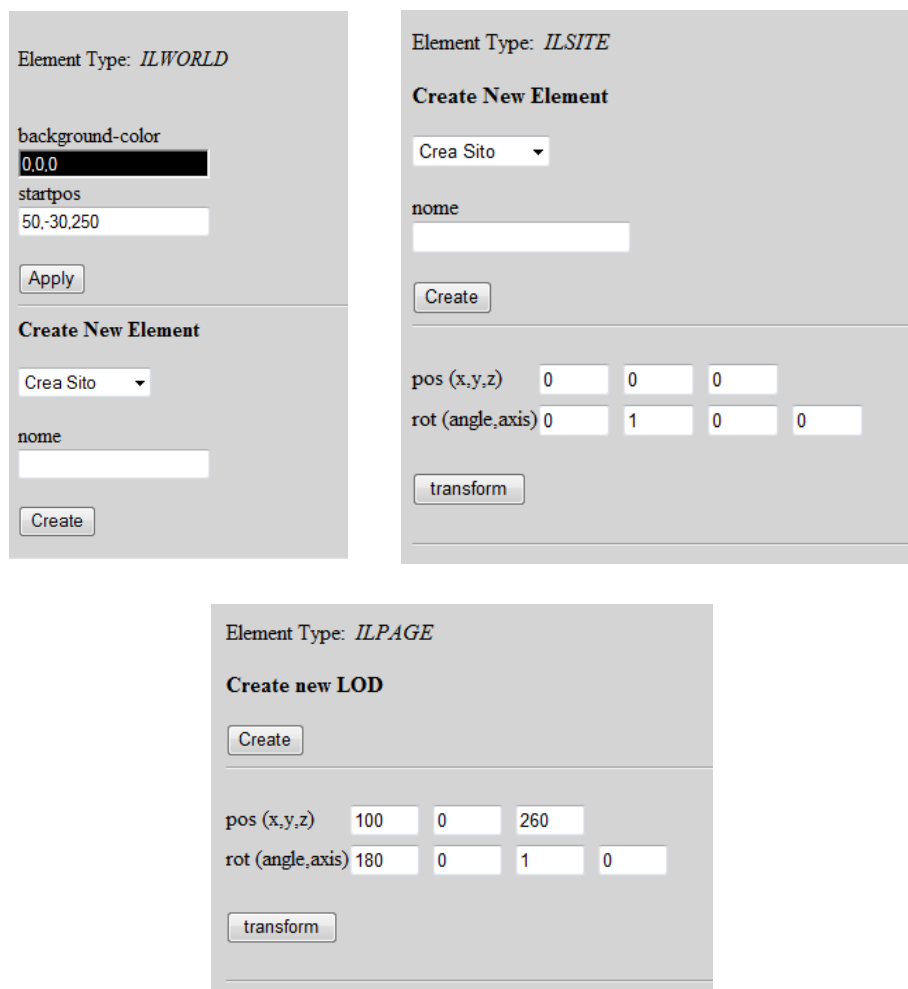
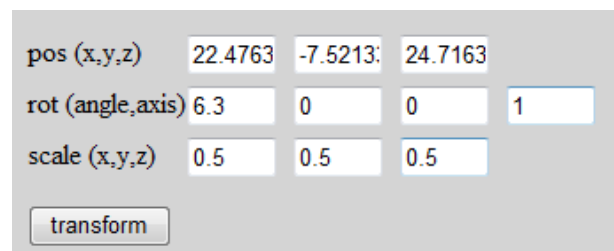


Figura 36 Pannelli di editing per la radice (ilworld), siti e pagine

Le opzioni consentono quindi di compiere varie operazioni: alcune consistono nella creazione di un oggetto figlio, altre invece permettono di modificare alcune proprietà dell'oggetto (es. colore, font nel caso del testo). Si noti che per gli oggetti ai quali è possibile applicare trasformazioni (traslazioni, scalature, rotazioni), è visualizzato inoltre un pannello che consente di impostare valori numerici di trasformazione (Figura 37), e che viene tenuto aggiornato durante le modifiche effettuate tramite gizmo.



Il pannello di impostazione manuale delle trasformazioni è un'interfaccia grafica con un sfondo grigio. Contiene tre righe di campi di input numerici e un pulsante. La prima riga, etichettata 'pos (x,y,z)', ha tre campi con i valori 22.4763, -7.5213 e 24.7163. La seconda riga, etichettata 'rot (angle,axis)', ha quattro campi con i valori 6.3, 0, 0 e 1. La terza riga, etichettata 'scale (x,y,z)', ha tre campi con i valori 0.5, 0.5 e 0.5. In basso a sinistra c'è un pulsante con la scritta 'transform'.

pos (x,y,z)	22.4763	-7.5213	24.7163	
rot (angle,axis)	6.3	0	0	1
scale (x,y,z)	0.5	0.5	0.5	

transform

Figura 37 Pannello di impostazione manuale delle trasformazioni

In caso di selezione della radice *world*, si potranno impostare alcune proprietà globali del paesaggio come il colore di sfondo o la posizione iniziale della camera. Inoltre è possibile scegliere se creare un sito o una pagina, specificarne il nome, quindi confermarne la creazione tramite il bottone *Create*.

In caso di selezione di un elemento di tipo *ILSITE* sarà possibile scegliere di creare un nuovo sito o una nuova pagina figlia del sito corrente.

In caso di selezione di un elemento di tipo *ILPAGE*, sarà mostrata l'interfaccia di creazione di un nuovo livello di dettaglio.

In caso di selezione di un elemento *ILLEVEL* (livello di dettaglio), verrà mostrata un'interfaccia che consente di scegliere se creare un oggetto testuale, un immagine, o un modello 3D (Figura 38).

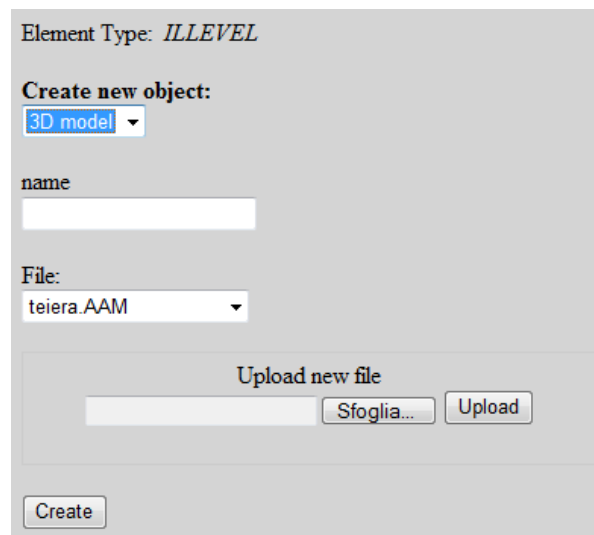
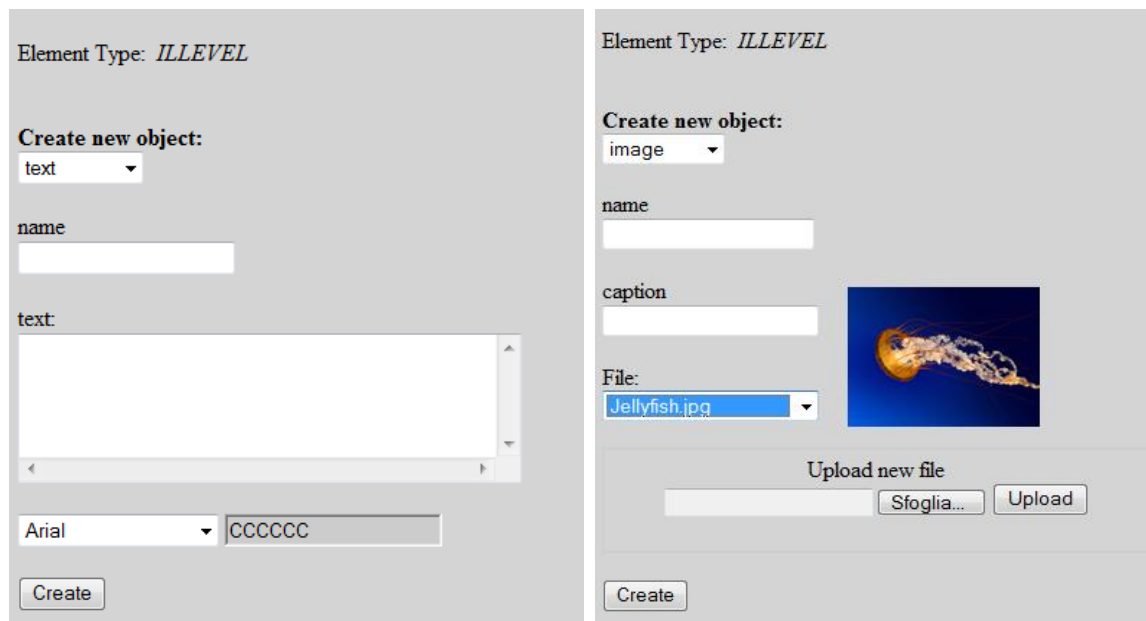


Figura 38 Pannelli di creazione di elementi testuali, immagini, modelli 3D

Se la scelta ricade su un elemento testuale, sarà possibile specificarne il nome, il contenuto testuale, il font da utilizzare, il colore.

Se la scelta ricade su un'immagine, sarà possibile specificarne il nome, una descrizione (utile, come vedremo, in fase di ricerca), scegliere un'immagine tra quelle disponibili sul server. I formati immagine accettati sono JPG e PNG. Per agevolare la scelta dell'immagine è presente una preview dell'immagine. Nel caso l'immagine che si vuole inserire non sia presente nell'elenco è possibile fare l'upload di un nuovo file sul server attraverso l'apposito modulo. L'immagine caricata sarà quindi disponibile nella lista dei file.

Quanto detto per le immagini vale anche per i modelli 3D con la differenza che non si fornisce un preview dei modelli 3D. Per il momento sono accettati solamente modelli nel formato di XVR e cioè AAM. Sono disponibili, in forma gratuita, opportuni esportatori di file AAM per i principali programmi di modellazione¹⁶.

In caso di selezione di un oggetto testuale, di un'immagine o di un modello 3D, interfacce simili alle precedenti saranno visualizzate per consentire la modifica delle proprietà.

3.2.8 L'interfaccia di caricamento, salvataggio e cancellazione di un IL

Al di sopra della 3D view troviamo l'interfaccia per l'apertura di un paesaggio di informazione disponibile sul server web (Figura 39). Una select-box consente la scelta tra gli ILs disponibili, mentre la pressione del bottone LoadIL determina il caricamento del nuovo paesaggio di informazione.



Figura 36 Pannello di caricamento, cancellazione e salvataggio di IL

Per poter conservare il paesaggio di Informazione creato, l'utente può sfruttare la funzionalità di salvataggio che l'applicazione mette a disposizione. L'IL sarà salvato sul server in una cartella zip, e sarà inoltre aggiunto nella lista degli ILs disponibili per il caricamento. La cartella zip, oltre a contenere il file ilxb che descrive la struttura del paesaggio (e che per convenzione ha lo stesso nome della cartella zip), contiene anche tutte le risorse (immagini, modelli 3D, ecc) necessarie.

È possibile eliminare un IL selezionandolo nella lista in alto a sinistra e poi premendo il bottone *deleteIL*. Ovviamente non è possibile eliminare il paesaggio su cui si sta attualmente lavorando.

3.2.9 L'Interfaccia di ricerca

Per agevolare l'utente nella ricerca di una particolare informazione si è dotato il software della funzionalità di ricerca testuale (Figura).

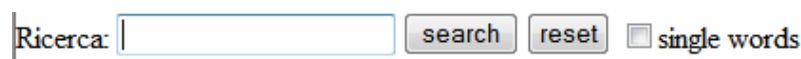


Figura 40 interfaccia per la ricerca testuale

¹⁶ http://wiki.vrmedia.it/index.php?title=Main_Page

L'utente deve cioè poter prendere velocemente visione di quali sono gli oggetti testuali che contengono l'informazione desiderata. La ricerca è stata estesa anche alle immagini, dotando tali oggetti di una proprietà descrizione (*caption*).

Nella soluzione adottata, l'utente deve digitare il testo che intende ricercare all'interno del campo di ricerca che si trova sotto il visualizzatore e poi premere il bottone search.

Esistono due modalità di ricerca. Nella prima modalità gli oggetti del paesaggio che contengono o soddisfano l'espressione digitata vengono evidenziati nel loro complesso all'interno della visualizzazione utilizzando un colore verde. Ammettiamo l'utente abbia digitato la parola *mezzo*, il risultato sarà il seguente (Figura 37):

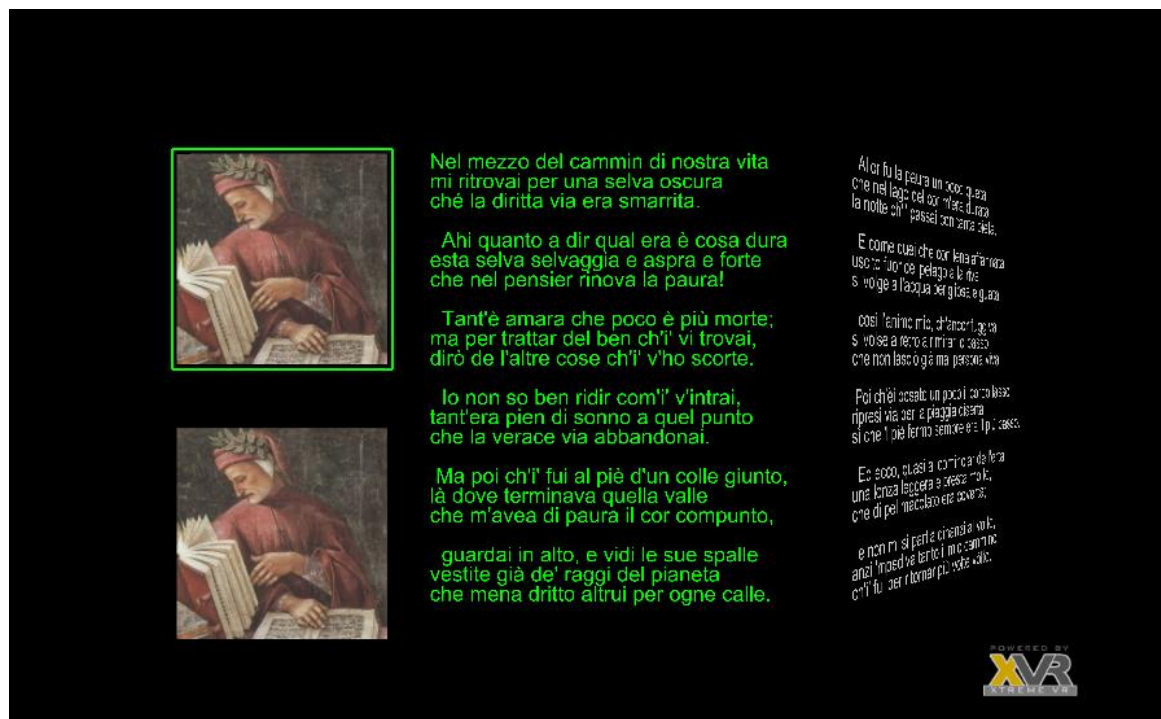
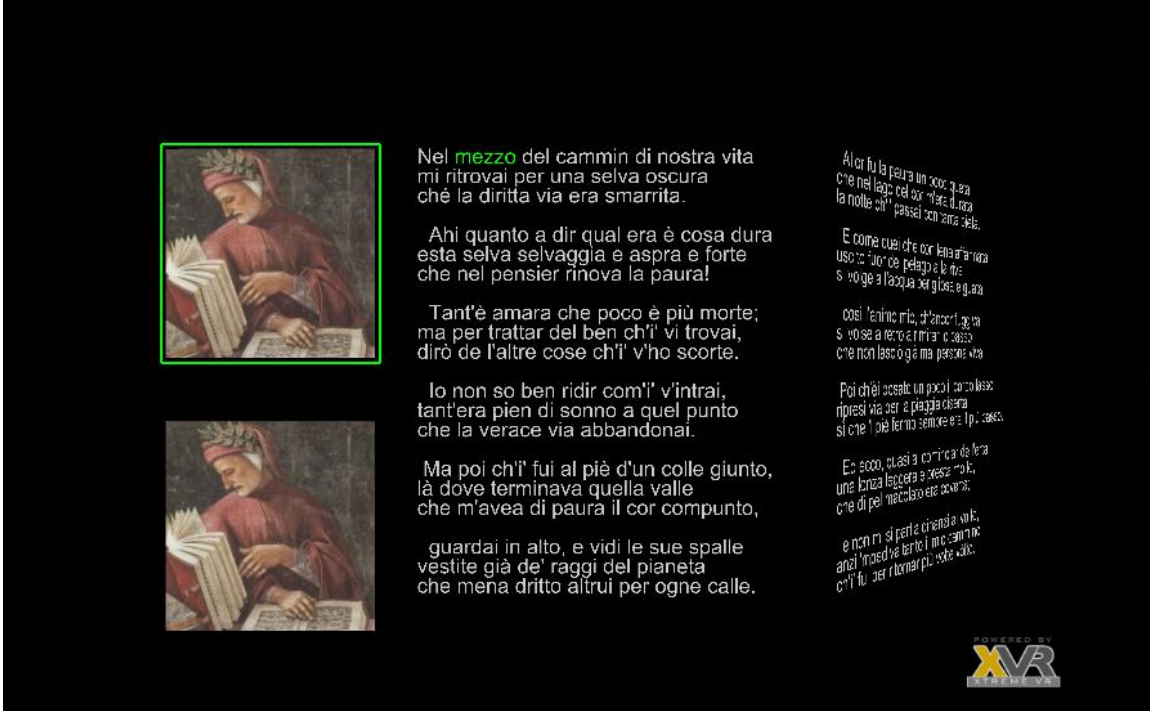


Figura 37 Esempio di risultato della ricerca della parola "mezzo"

Come si pu  osservare si   evidenziato il primo blocco di testo dato che contiene *mezzo* come seconda parola; anche l'immagine in alto   evidenziata in quanto nella sua didascalia (*caption*) conteneva tale parola.

Selezionando per  la check-box *single words* attiviamo la seconda modalit  di ricerca, che prevede che siano evidenziate solamente le singole parole. Ammettendo la stessa ricerca precedente otterremmo il seguente risultato (Figura 38):



Nel **mezzo** del cammin di nostra vita
mi ritrovai per una selva oscura
ché la diritta via era smarrita.

Ahi quanto a dir qual era è cosa dura
esta selva selvaggia e aspra e forte
che nel pensier rinova la paura!

Tant'è amara che poco è più morte;
ma per trattar del ben ch'i' vi trovai,
dirò de l'altre cose ch'i' v'ho scorte.

Io non so ben ridir com'i' v'intrai,
tant'era pien di sonno a quel punto
che la verace via abbandonai.

Ma poi ch'i' fui al piè d'un colle giunto,
là dove terminava quella valle
che m'avea di paura il cor compunto,

guardai in alto, e vidi le sue spalle
vestite già de' raggi del pianeta
che mena dritto altrui per ogni calle.

Alor fu la paura un poco quassa
che nel lago del cor m'era durata
la notte ch'i' passai con tanta allegria.

E come quei che con lena affaticata
uscio fuor del pelago a la riva
si volge a l'acqua del gioioso guata.

Così l'animo mio, sì ben disposto
si mosse a retro e rimira: e vobis
che non lascio già ma persona viva.

Poi ch'èi accorto un poco del mio essere
ripresi via per la piaggia diserta
sì che 'l piè fermo sempre era 'l piè cascato.

Ed ecco, quasi a cominciar del monte
una brezza leggera a piegar tutto
che di pel miccolato era costato.

E non m' si partì ch'andava vultoso
anzi 'l muso ve l'avea il mio cammino
ch'i' fu per ritornar più volte vòto.

POWERED BY
XVR

Figura 38 Esempio della ricerca della parola mezzo in modalità di ricerca single-words

All'interno del riquadro testuale è dunque possibile inserire una singola parola o anche più parole. In caso di ricerca di più parole, il motore effettuerà implicitamente un AND tra le due parole: la query "Ciao Mondo" sarà interpretata dal motore di ricerca come "Ciao AND Mondo" per cui si restituiranno tutti i documenti che contengono contemporaneamente la parola *Ciao* e la parola *Mondo*.

Oltre alla ricerca di singole parole o frasi, è possibile effettuare ricerche speciali. È possibile, per esempio, richiedere tutti i documenti che contengono parole che iniziano in un certo modo utilizzando l'operatore *. Per esempio se cercassimo tutte le parole che iniziano per p scriveremmo p* ottenendo, in caso di visualizzazione single-words il seguente risultato (Figura 39)

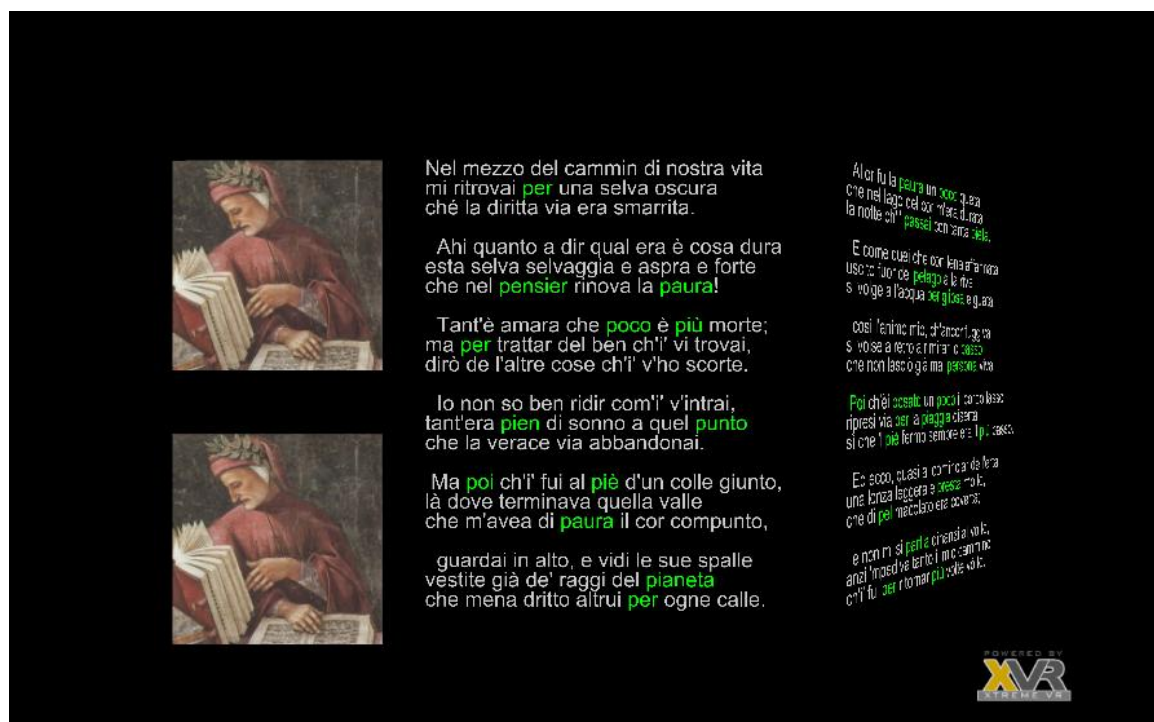


Figura 39 Esempio di ricerca di tutte le parole che iniziano per "p"

È possibile inoltre richiedere tutti i documenti che contengono una parola ma non un'altra con l'operatore "-" (`word1 - word2`) oppure tutti i documenti che contengono una parola oppure un'altra con l'operatore OR (`word1 OR word2`).

E' doveroso specificare che nell'attuale implementazione le due tipologie di ricerche potrebbero dare risultati leggermente diversi. Nella prima modalità la ricerca avviene su una tabella che memorizza in ogni record l'intero testo di ciascun elemento, mentre nella seconda modalità vengono memorizzate separatamente le singole righe di testo di ogni elemento¹⁷. Quindi nel primo caso la ricerca avviene per elemento, nel secondo caso avviene per riga.

¹⁷ La ragione sta nel fatto che il sistema di rendering deve ricevere i risultati organizzati per singola riga, e il metodo più veloce e semplice di ottenere questo è stato quello adottato. Un miglioramento futuro sarà quello di effettuare, anche per la ricerca per parole, una ricerca su una tabella organizzata per elementi e poi trasformare i risultati ottenuti in modo che si riferiscano ad ogni singola riga.

3.3 Descrizione dell'implementazione

L'implementazione di funzionalità di editing per i paesaggi di Informazione accessibili dal web è stata effettuata a partire dal software di visualizzazione di IL sviluppato in XVR che abbiamo descritto nel dettaglio nel capitolo precedente.

L'applicazione è stata pensata in un ottica client-server. Perché l'applicazione funzioni è dunque necessario installarla su un server web che supporti il linguaggio di scripting PHP. Nello sviluppo si è utilizzato l'ambiente EasyPHP.

L'applicazione consiste in una cartella contenente una pagina web, il bytecode dell'applicazione e alcune risorse. In particolare la sottocartella ILs contiene tutti gli ILs salvati (oltre a quello di default), la sottocartella upload contiene le risorse (immagini, modelli 3D) caricati sul server. In più sono presenti alcuni file aggiuntivi (script, elementi grafici, ecc.). L'applicazione è accessibile da browser web Internet Explorer all'URL dove essa è stata installata.

In linea generale, lo sviluppo delle funzionalità di editing è consistita in primo luogo nella creazione di un'interfaccia grafica web che permettesse all'utente di selezionare l'operazione di editing desiderata. Opportuni gestori di eventi relativi all'interfaccia, sviluppati nel linguaggio di scripting Javascript, si occupano invece dell'invio di comandi all'applicazione XVR. Analogamente, se necessario, il programma XVR potrà comunicare con la pagina web inviando opportuni comandi o informazioni alla pagina web.

Questo è possibile in quanto la tecnologia XVR mette a disposizione funzioni (DataIn() e DataOut()) per lo scambio di stringhe di caratteri tra l'applicazione XVR e la pagina HTML che la contiene e viceversa.

La comunicazione tra la pagina Web e il server (per esempio per le operazioni di salvataggio, o per ottenere liste delle risorse presenti sul server) è gestita tramite richieste AJAX.

La seguente illustrazione (Figura 40) schematizza l'architettura complessiva del software.

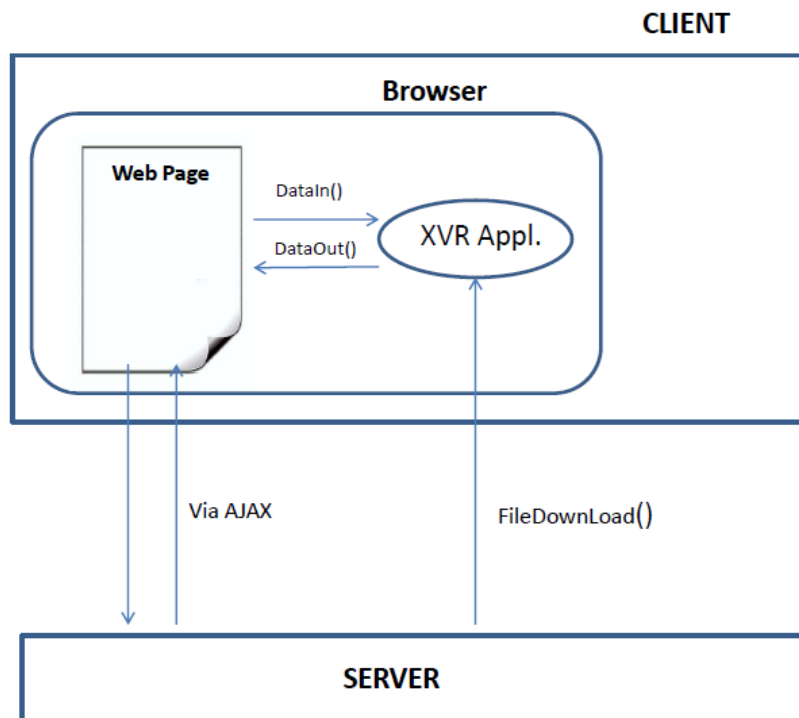


Figura 40 Schema dell'architettura dell'applicazione

Nei seguenti paragrafi descriveremo le singole funzionalità sviluppate, analizzando i problemi, motivando le scelte progettuali e implementative.

3.3.1 Tecnologie e linguaggi utilizzati

Le varie funzionalità sono state realizzate utilizzando le seguenti tecnologie e linguaggi come: XVR, OpenGL, Javascript, PHP, HTML, CSS.

XVR è il framework utilizzato per la visualizzazione dell'ambiente 3D e per l'implementazione di funzionalità interne all'ambiente. La sua struttura è già stata spiegata nel capitolo 2. Per la realizzazione di alcune funzionalità specifiche (come il disegno dei bbox, il disegno dei gizmos ecc) le funzioni messe a disposizione da XVR non bastavano, per cui è stato necessario ricorrere ad una libreria di più basso livello (OpenGL).

OpenGL (Open Graphics Library) è una collezione di API multilinguaggio e multiplatforma per applicazioni orientate alla computer grafica 2D e 3D. Essa astrae la programmazione dallo specifico hardware grafico utilizzato, offrendo al programmatore una unica API. In generale, il compito di OpenGL è quello di ricevere primitive come punti, linee e poligoni e convertirle in pixel per la visualizzazione a schermo.

Javascript è un linguaggio di scripting lato client, che viene incorporato all'interno di una pagina web allo scopo di aggiungere comportamenti e interazioni con l'utente. Si tratta di un linguaggio di programmazione orientato agli oggetti e interpretato dalla maggior parte dei browser web in circolazione. Si è utilizzato Javascript allo scopo di gestire gli eventi dovuti all'interazione dell'utente ed inviare comandi all'applicazione XVR. Si è inoltre fatto ricorso a Javascript per lo scambio di dati in modalità asincrona con il server (tecnica AJAX).

PHP è un linguaggio di programmazione interpretato utilizzato principalmente per sviluppare applicazioni web lato server. Nel progetto descritto viene utilizzato soprattutto per rispondere alle chiamate AJAX inviate dal client per il salvataggio o rimozioni di IL dal server o per il recupero di risorse (immagini, modelli 3d) dal server.

HTML (Hypertext Markup Language) è il linguaggio di marcatura utilizzato nei documenti ipertestuali nel World Wide Web. Esso è stato utilizzato nella creazione della struttura dell'interfaccia di editing.

CSS è un linguaggio utilizzato per definire la formattazione di documenti HTML o XML. Esso è stato adottato all'interno del software per la formattazione dell'interfaccia. Si sono utilizzati per lo più stili incorporati direttamente nella pagina (embedded CSS) o nei tag degli elementi (inline CSS) dato che essendo l'applicazione formata da una sola pagina non era fondamentale creare un foglio di stile esterno.

3.3.2 Definizione di un protocollo per lo scambio dei dati da XVR a pagina web e viceversa

Perché sia possibile lo scambio di comandi dalla pagina web all'applicazione XVR occorre per prima cosa definire un protocollo di scambio.

La tecnologia XVR mette a disposizione due funzioni per lo scambio di stringhe testuali tra la pagina HTML e l'applicazione in essa contenuta. In particolare, la funzione `DataIn()`, utilizzata all'interno di funzioni Javascript, permette alla pagina HTML di inviare messaggi, solitamente associati a particolari eventi come la pressione di link o bottoni.

Esiste poi la funzione `DataOut()` che permette all'applicazione XVR di inviare messaggi alla pagina HTML.

Come si può vedere, le funzioni `DataIn()` e `DataOut()` si prestano perfettamente ai nostri scopi. Da sole però non sono sufficienti a risolvere il problema, dato che esse di per sé inviano semplici stringhe testuali. Occorre definire un formato per lo scambio dei comandi.

Dato che un comando consiste in un nome e in uno o più parametri abbiamo pensato al formato JSON come una buona opzione per strutturare i messaggi da scambiare con l'applicazione. JSON (JavaScript Object Notation) è un formato di interscambio di dati in formato testuale. È basato sul linguaggio Javascript Standard ECMA-262 3^a edizione dicembre 1999, ma ne è indipendente. I tipi di dato supportati da questo formato sono:

- booleani (true e false);
- interi, reali, virgola mobile;
- stringhe racchiuse da doppi apici (" ");
- array (sequenze ordinate di valori, separati da virgole e racchiusi in parentesi quadre []);
- array associativi o oggetti (sequenze coppie chiave-valore separate da virgole racchiuse in parentesi graffe);
- null

L'uso di JSON tramite Javascript è particolarmente semplice, infatti è possibile eseguire il parsing di una stringa JSON tramite una semplice invocazione della funzione `eval()`. Possiamo utilizzare invece funzioni costruite ad-hoc per eseguire la conversione di un dato in una stringa in formato JSON.

Dal lato XVR, è possibile utilizzare una libreria di funzioni (`json.h.s3d`) per la gestione del formato JSON: nello specifico, la funzione `JSONparse(text)` si utilizza per il parsing di una stringa JSON mentre la funzione `JSONstringify(value)` si utilizza per la generazione di una stringa JSON a partire da un dato.

Nel nostro caso specifico, utilizzando JSON possiamo comodamente rappresentare un comando e i relativi parametri come un array e cioè come una sequenza di valori in cui il primo valore è il nome del comando e i successivi valori sono i parametri. Seguendo le

regole di sintassi stabilite dal formato, un comando potrà essere espresso tramite una stringa avente la seguente forma:

[<nome_comando>,<par1>, <par2>, ...]¹⁸

Partendo da queste basi, abbiamo dunque sviluppato il codice che si occupa della gestione dello scambio sia dal lato della pagina web sia dal lato dell'applicazione XVR.

Dal lato della pagina web¹⁹ è stata creata la funzione *manageButton(btn)* che gestisce gli eventi associati all'interazione dell'utente con i controlli dell'interfaccia: in base al controllo attivato, essa si occupa dell'invio dell'opportuno comando a XVR. Per motivi di spazio e di chiarezza riportiamo solamente una versione schematica della funzione:

```
function manageButton(btn)
{
    switch(btn.name)
    {
        /* A seconda del valore di btn.name (nome controllo attivato)
           si invia l'opportuno comando a XVR */
    }
}
```

Il significato e la sintassi dei possibili comandi sarà riportato nei paragrafi successivi.

Dal lato XVR²⁰ è stata definita la funzione *ReceiveDataFromWeb()* invocata dalla funzione *OnFrame()*. Essa riceve eventuali messaggi presenti, ne fa il parsing e in base al valore del primo parametro stabilisce di che comando si tratta ed esegue il codice relativo. Anche di questa funzione riportiamo una versione schematica.

```
function ReceiveDataFromWeb()
{
    var s = DataIn();
    if(s != Void && s != "" && s!=" ")
    {
        var p = JSONparse(s);
        switch(p[0])
        {
            /* In base al valore di p[0] si esegue un
               determinato codice di gestione del comando */
        }
    }
}
```

¹⁸ Per maggiore chiarezza nella tesi si utilizzano le parentesi angolari per indicare dei parametri. Esse non sono presenti nei reali comandi.

¹⁹ file *ilview.htm*

²⁰ file *ilview.s3d*

Per quanto riguarda l'invio dei dati dall'applicazione XVR alla pagina web, si utilizza semplicemente la funzione `DataOut()`.

I comandi sono ricevuti tramite il seguente script che riportiamo schematicamente:

```
<SCRIPT LANGUAGE="JavaScript" FOR="xvr" EVENT="DataOut(s)">
{
    var cmd=eval(s);
    switch(cmd[0])
    {
        /* A seconda del tipo di comando ricevuto viene
        eseguito il codice corrispondente */
    }
}
```

3.3.3 Modalità di editing, modalità di viewer

Dato che lo scopo era ottenere un software che fungesse sia da visualizzatore che da editor, si è deciso di separare le due modalità. In modalità editor (modalità impostata di default all'apertura della pagina web), viene visualizzata un'apposita interfaccia di editing; inoltre, l'interazione e la visualizzazione della vista 3D è diversa rispetto a quella presente nella modalità viewer, in quanto pensata appositamente per agevolare le funzionalità di editing. Per esempio, la modalità di editing supporta la visualizzazione dei bbox per gli elementi selezionati, la visualizzazione dei gizmo per la manipolazione degli elementi e altre funzionalità.

Il passaggio da una modalità all'altra avviene semplicemente selezionando o deselegionando la checkbox *viewer mode* presente sopra la vista 3D. Ogni volta che si modifica lo stato della checkbox viene inviato all'applicazione XVR il seguente comando:

```
["mode", <par>]
```

dove <par> vale *true* se si vuole attivare la modalità *viewer*, *false* se si desidera la modalità di editing.

All'interno di XVR si tiene traccia della modalità selezionata, aggiornando la variabile globale *mode*²¹.

²¹ La variabile *mode* contiene la stringa *"editor"* per la modalità editor, mentre contiene la stringa *"viewer"* per la modalità di visualizzazione. Di default è impostata a *"editor"*.

3.3.4 Creazione di un indice degli elementi dell'IL: la vista di struttura

La vista di struttura è stata realizzata tramite una select-box: i vari elementi sono riportati come *options* con il nome che è stato dato loro in fase di creazione preceduto dal nome della loro classe di appartenenza, mentre la struttura è espressa mediante un'opportuna indentazione.

Si è dovuto predisporre l'applicazione XVR in modo che all'avvio venga inviato alla pagina web un messaggio contenente la struttura dell'IL, mentre la pagina web tramite un opportuno script dovrà occuparsi di riempire la select-box a partire dalla struttura descritta dal messaggio.

Per fare questo, lato XVR è stata creata la funzione *UpdateJSONStructure(elem)*, la quale si occupa di visitare ricorsivamente tutta la struttura ad albero che rappresenta il paesaggio di informazione attualmente visualizzato e aggiorna alcune variabili con le informazioni necessarie a ricostruire la struttura sulla pagina web e compiere operazioni. In particolare sono create tre stringhe testuali in formato JSON:

- structure: rappresenta la struttura del nostro IL nella forma [nome, [figli]]
- pathstructure: memorizza la struttura del nostro IL nella forma [path, [figli]]. Per path intendiamo un percorso che identifica univocamente un elemento all'interno della gerarchia dell'IL. I path si esprimono a partire dalla radice (che è identificata da /) riportando i vari elementi presenti nel cammino che va dalla radice all'elemento considerato separati dal carattere / (per esempio /sito/pagina1/0/testo)²². Dal lato della pagina web è importante avere una conoscenza del path dei vari elementi in quanto questi ci potranno servire come parametri dei comandi da inviare all'applicazione XVR .
- typestructure: memorizza la struttura del nostro IL nella forma [type, [figli]]. Per type intendiamo il nome della classe di appartenenza di un elemento per esempio ILWorld, ILSite, ILPage ecc.. Ci siamo occupati di memorizzare anche queste informazioni perché si potrebbero rivelare utili nella programmazione di alcuni comportamenti all'interno dello script javascript.

Il comando per l'invio delle informazioni di struttura alla pagina web è una stringa JSON avente la seguente struttura:

²² Nei path non è obbligatorio inserire il livello di dettaglio. Pertanto, i path *"/sito/pagina1/0/testo"* e *"/sito/pagina1/testo"* sono interpretati in modo equivalente.

["structure", <structure>, <pathstructure>, <typestructure>]

dove il primo elemento è la stringa "structure" che serve a indicare il comando, mentre gli altri elementi sono le stringhe JSON determinate sopra.

Dal lato della pagina Web, il codice javascript di ricezione comandi riconoscerà il comando e si occuperà di invocare la funzione *createStructure(elementi,path,type)* per l'inizializzazione della selectbox struttura. In particolare il tipo e il nome di ciascun elemento sarà inserito come nodo testuale della option e opportunamente indentato, l'attributo name conterrà il tipo di oggetto , e l'attributo value conterrà il path.

3.3.5 Selezione di un elemento

Come specificato sopra, la selezione di un oggetto può avvenire facendo doppio clic su di esso all'interno della *3D View* oppure della vista struttura.

Nel caso di selezione effettuata all'interno della finestra 3D, è necessario che l'applicazione XVR invii alla pagina web l'informazione su quale elemento è stato selezionato così che possa mantenere sincronizzata anche la Vista di Struttura e visualizzare l'opportuna interfaccia di editing per quell'elemento. Per fare questo, nel momento in cui viene rilevata una selezione, l'applicazione XVR invia il seguente comando JSON alla pagina web:

["sel", <path>]

dove <path> è il percorso dell'oggetto selezionato. Il codice javascript una volta riconosciuto il comando, aggiorna opportunamente la vista di struttura e l'interfaccia di editing.

Nel caso in cui la selezione avvenga tramite la Vista di Struttura, la pagina web invia all'applicazione XVR un comando avente lo stesso formato del precedente, in modo da informare l'applicazione XVR di quale sia l'oggetto selezionato²³.

Al fine di poter riempire i campi di editing dell'elemento selezionato, l'applicazione XVR invia automaticamente alla pagina web le informazioni necessarie per mezzo dei seguenti comandi²⁴:

²³ L'applicazione tiene traccia dell'oggetto attualmente selezionato all'interno della variabile globale SelectedElem.

²⁴ Descrizione dettagliata di ciascun comando in appendice A

- ["text_info", <text>, , <color>, <pos>, <scale>] per gli oggetti testuali
- ["image_info", <caption>, <filename> <pos>, <size>] per le immagini
- ["model_info", <filename>, <pos>, <scale>, <orient>] per i modelli 3D
- ["page_info", <pos>, <orient>] per gli oggetti pagina
- ["site_info", <pos>, <orient>] per gli oggetti sito
- ["world_info", <startPos>, <backgroundColor>] per l'oggetto radice

La gestione della selezione in modalità editor (diversa modalità di interazione, richiesta del disegno dei bbox della selezione ecc.) è stata ottenuta ampliando il codice della funzione ILWorld::ProcessSelection().

3.3.6 Creazione e cancellazione di elementi

Una delle funzionalità di base che un editor deve fornire, è la creazione e la cancellazione di elementi. Entrambe le operazioni sono state pensate per essere effettuate selezionando l'elemento interessato dall'operazione.

Nel caso della cancellazione si utilizza il bottone *delete*. Il bottone premuto sarà riconosciuto dal gestore di eventi javascript che invierà a XVR un comando avente struttura:

```
['remove', <path>]
```

dove path è il percorso identificativo dell'elemento che intendiamo cancellare. L'applicazione XVR, ricevuto il comando, esegue il codice di cancellazione. A cancellazione avvenuta sarà inviata la nuova struttura dell'IL alla pagina web per l'aggiornamento della Vista di Struttura.

Invece, nel caso in cui l'obiettivo sia la creazione di un nuovo elemento, si selezionerà l'elemento del quale vogliamo creare il figlio. A seconda della tipologia dell'elemento selezionato, il gestore di eventi fa comparire una diversa interfaccia di creazione di nuovi elementi nel Pannello di Editing. L'interfaccia sarà costituita da vari campi per l'input. In

particolare il campo per l'input del colore del testo è creato e gestito tramite una libreria javascript (JSColor) esterna incorporata nel file principale²⁵.

Una volta premuto il bottone *Crea*, verrà inviato un comando a XVR che avrà le seguenti forme a seconda dell'elemento che si sta creando:

- ['new', <ctx>, 'ILLEVEL'] per creare un nuovo livello di dettaglio
- ['new', <ctx>, 'ILPAGE', <name>] per creare una nuova pagina
- ['new', <ctx>, 'ILSITE', <name>] per creare un nuovo sito
- ['new', <ctx>, 'ILTEXT', <name>, <text>] per creare un nuovo testo
- ['new', <ctx>, 'ILIMAGE', <name>, <url>] per creare una nuova immagine
- ['new', <ctx>, 'ILIMODEL', <name>, <url>] per creare un nuovo modello 3D

dove 'new' indica il comando, <ctx> indica il path dell'oggetto di cui vogliamo creare un figlio (in pratica si tratta dell'elemento selezionato), il terzo parametro indica il tipo di oggetto che si vuole creare, mentre i successivi parametri sono specifici del particolare tipo di oggetto creato²⁶. L'applicazione XVR, ricevuto il comando, esegue il codice di creazione dell'elemento specificato.

Merita un'attenzione particolare il caso della creazione di un oggetto immagine o di un modello 3D. In entrambi i casi è necessario infatti effettuare un trasferimento (upload) della risorsa (immagine, modello 3D) sul server web allo scopo di poter essere successivamente caricata in modo asincrono dall'applicazione XVR.

Per fare questo nel form per la creazione di un elemento immagine è stato inserito un modulo di upload di file che invia i dati ad uno script php *upload.php*:

```
<form class="load" action="upload_image.php" method="post"
enctype="multipart/form-data" target="upload_target"
onsubmit="manageButton(this) >
    File: <input name="myfile" type="file" />
        <input type="submit" name="submitBtn" value="Upload" />
</form>
```

²⁵ JSColor è rilasciato nel sito <http://jscolor.com/> con licenza LGPL ed è perciò utilizzabile in qualunque applicazione anche commerciale.

²⁶ Per informazioni dettagliate vedere l'appendice A

```
<iframe id="upload_target" name="upload_target" src="#"
style="width:0;height:0;border:0px solid #fff;">
</iframe>
```

Come si può vedere, poiché vogliamo che l'invio del file avvenga senza abbandonare o ricaricare la pagina contenente l'applicazione, è stato utilizzato un iframe nascosto, e si è settata la proprietà target del form di invio al nome di questo iframe. In questo modo, al momento dell'invio del file, la pagina *upload_image.php* si aprirà all'interno dell'iframe lasciando inalterata la pagina principale. Lo script php memorizza il file ricevuto nella cartella *upload*²⁷ del server. In coda allo script php per la memorizzazione del file sul server, è stato inserito uno script javascript che richiama la funzione *stopUploadImage()* la quale si occupa di restituire un messaggio di successo o insuccesso nel caricamento del file e di bloccare un animazione mostrata durante il caricamento²⁸.

Successivamente, la pressione del bottone *Crea* determinerà l'invio del comando di creazione dell'immagine. Dal lato dell'applicazione XVR, prima della creazione del nuovo elemento è necessario effettuare il caricamento asincrono del file. A tal scopo si utilizza la funzione *FileDownload()* che richiede il caricamento asincrono della risorsa e restituisce un id unico. A completamento del caricamento viene automaticamente chiamata la funzione *DownloadReady()* all'interno della quale si procede alla creazione dell'elemento e all'aggiunta di esso nel paesaggio di informazione.

Un procedimento analogo è stato applicato all'aggiunta di modelli 3D all'interno dell'ambiente.

Modifica di elementi

Se lo si desidera, è possibile modificare oggetti già esistenti. Per esempio selezionando un oggetto testuale sarà mostrata un'opportuna interfaccia per la modifica del contenuto del testo, del carattere, del colore. Selezionando un'oggetto immagine, sarà possibile sostituire l'immagine e la didascalia. Selezionando un modello 3D sarà possibile

²⁷ Più precisamente le immagini sono inserite nella sottocartella *images*, mentre i modelli 3D in quella denominata *3d_models*

²⁸ L'animazione viene attivata al momento del caricamento da parte del gestore dell'evento *onsubmit*.


sostituirlo con un altro modello 3D. La modifica si applica premendo il bottone *Modify* che ha come effetto l'invio di uno dei seguenti comandi, a seconda del tipo di oggetto su cui operiamo:

- ['modify', <path>, <text>, , <color>] per oggetti testuali
- ['modify', <path>, <url>, <caption>] per oggetti immagine
- ['modify', <path>, <url>] per modelli 3D

Per il settaggio delle proprietà globali della scena, come il colore o la posizione iniziale di camera, si utilizza invece il comando:

```
['global_prop', <color>, <startPos>]
```

Creazione di un link tra due oggetti

Tra due oggetti può essere creato un link. Per fare questo, dopo aver selezionato l'elemento sorgente e destinazione, è possibile cliccare sul bottone . In questo modo viene inviato il seguente comando a XVR:

```
['newlink']
```

Esso ha come effetto la creazione di un link e la sua visualizzazione.

Si noti che non sono necessari parametri, dato che l'applicazione XVR tiene traccia dei due oggetti selezionati²⁹.

Per impostare i parametri di un link è necessario fare click sul link. In questo modo l'applicazione XVR invia alla pagina il comando:

```
['link_info', <degree>, <viewpoint>, <time>]
```

che ha l'effetto di far apparire nel pannello di editing i campi che consentono di impostare il grado della curva, la posizione in cui si dovrà trovare la camera alla fine del percorso³⁰, il tempo di percorrenza del link in secondi.

Premendo il pulsante Apply viene inviato ad XVR il comando:

```
['linkmodify', <degree>, <viewpoint>, <time>]
```

²⁹ In particolare tali informazioni sono contenute nelle variabili globali SelectedElem e SelectedElem2.

³⁰ Tale posizione è espressa rispetto all'oggetto destinazione.

che richiede l'applicazione delle modifiche effettuate. Premendo invece il pulsante LinkPreview viene inviato il comando

```
['linkpreview']
```

che porta la camera a posizionarsi nel punto di destinazione specificato per il link, consentendoci di valutare in anteprima la necessità di effettuare altre modifiche.

3.3.7 Manipolazione degli elementi: traslazione, scalatura, rotazione

Il disegno dei manipolatori di traslazione, rotazione e scalatura per l'oggetto selezionato viene attivato o disattivato agendo sul checkbox *Show object gizmo*. Tale azione ha infatti come effetto l'invio all'applicazione XVR del comando:

```
['showGizmos', <par1>]
```

dove <par1> può valere true (visualizzazione attivata) o false. Nell'applicazione XVR tale valore sarà inserito nella variabile globale optShowGizmo.

Di default viene visualizzato il gizmo di traslazione, ma è possibile sceglierne un'altro tra quelli disponibili nella select box *Edit Mode*, azione che ha come effetto l'invio del comando:

```
['changeGizmoType', <gizmoType>]
```

dove <gizmoType> può valere "translate", "scale", "rotate" o "uniformscale".

Il disegno dei manipolatori avviene tramite codice OpenGL contenuto nelle funzioni *ILWorld::DrawGizmo(elem)* per il gizmo di traslazione e di scalatura e *ILWorld::DrawArcBall(elem)* per quello di rotazione.

Un problema che è stato necessario affrontare e che riguarda tutti e tre i tipi di manipolatori è stato quello di trovare il modo di effettuare una selezione delle maniglie messe a disposizione da tali manipolatori. Come già sappiamo, il viewer utilizza per il rilevamento della selezione di oggetti la tecnica della selezione per colore (vedi capitolo 2) : ai diversi oggetti visibili della scena viene assegnato un colore dalla funzione *nextSelectiond()* della classe *ILWord*, inserendo tali oggetti all'interno dell'array *SelectionObject* e utilizzando il valore di posizione come colore per il rendering.

Per rendere selezionabili le maniglie che costituiscono i gizmos è stato necessario quindi riadattare per i nostri scopi il meccanismo di selezione esistente. Quando viene attivata la visualizzazione dei gizmo, nell'array `SelectionObject`, insieme agli oggetti visibili nella scena, inseriamo altri tre oggetti, tre stringhe ("GizmoX", "GizmoY", "GizmoZ") una per ogni maniglia di cui è composto il gizmo. Le maniglie durante il rendering di selezione verranno disegnate del colore specificato dalla posizione di ciascuna delle stringhe nell'array. A questo punto, quando nella funzione `ILWorld::processSelection()` viene rilevata una selezione, se tale oggetto è una stringa siamo certi di aver selezionato un gizmo, e il contenuto della stringa ci permetterà di capire la precisa maniglia che è stata selezionata.

La funzione di gestione dei gizmos (`ILWorld::ProcessGizmo(GizmoSel)`), invocata per tutta la durata dell'interazione con il gizmo, una volta stabilito il tipo di gizmo attivato e quale maniglia è stata selezionata, attuerà un determinato comportamento al fine di applicare le trasformazioni desiderate.

Descrizione implementazione gizmo di traslazione e scalatura

Perché il manipolatore di traslazione funzioni correttamente non è sufficiente eseguire una differenza tra la posizione corrente del mouse e quella precedente, quindi sommare tale differenza alla posizione corrente dell'oggetto. Facendo in questo modo infatti l'oggetto non si sposterebbe della quantità espressa dalla traslazione del mouse, determinando problemi di usabilità per l'utente. Questo perché in grafica 3D per una stessa scena esistono numerosi sistemi di coordinate. Abbiamo il sistema di *riferimento locale*, cioè specifico di ogni oggetto, abbiamo le *coordinate mondo* cioè espresse secondo un sistema di riferimento globale della scena, abbiamo le *coordinate di vista* cioè espresse prendendo come assi di riferimento quelli della camera, abbiamo le *coordinate di window* (il risultato dell'operazione di proiezione eseguita durante il rendering per trasformare la scena da 3D a 2D), le *coordinate di viewport* ovvero le coordinate nella finestra dello schermo su cui è renderizzata la scena, e infine le *coordinate del mouse*.

Perché l'entità dello spostamento dell'oggetto corrisponda a quella espressa dallo spostamento del mouse occorre quindi prendere i punti che esprimono inizio e fine dello spostamento in coordinate mouse e trasformarli in punti nello spazio tridimensionale.

Per fare questo abbiamo fatto ricorso ad un algoritmo che determina l'istante (o tempo) di distanza minima tra due segmenti³¹. I segmenti considerati sono l'asse del gizmo su cui si vuole effettuare lo spostamento e il raggio del mouse (determinato congiungendo il punto del mouse sul piano near e sul piano far del view frustum). Sostituendo l'istante di tempo determinato in un'equazione parametrica che esprime l'asse selezionato riusciamo a determinare, con buona approssimazione, le coordinate 3d corrispondenti al punto espresso in coordinate mouse. Questo procedimento va ripetuto per i due punti che costituiscono l'inizio e la fine del drag con il mouse. Trasformando i punti trovati in coordinate locali al padre dell'oggetto su cui operiamo la trasformazione, effettuando una differenza tra tali punti e sommando tale valore alla posizione dell'oggetto, otteniamo un corretto funzionamento del gizmo di traslazione.

Una soluzione simile è stata adottata per il gizmo di scalatura. La nuova scala viene calcolata in funzione della differenza tra punto finale e punto iniziale del drag. Per quanto riguarda la scalatura è inoltre importante osservare che, per gli oggetti ILTEXT sono possibili solo scalature uniformi mentre per gli altri oggetti è possibile scalare anche solo su un asse specificato.

Descrizione implementazione gizmo di rotazione

Per il gizmo di rotazione una soluzione standard descritta in letteratura è quella denominata Arcball (Shoemake, 92). Tuttavia per il progetto si è utilizzata una soluzione più semplice da implementare. L'utente seleziona con un click il cerchio che rappresenta una direzione di rotazione, quindi sposta il mouse orizzontalmente per applicare la rotazione. L'entità della rotazione nella direzione selezionata è determinata calcolando una differenza (un delta) tra la posizione corrente del mouse e quella precedente, il tutto moltiplicato per un fattore di scala. Per esempio, in caso di rotazione sull'asse x essa corrisponderà a:

[1,0,0,delta_x*0.1]

La rotazione determinata deve essere combinata all'orientazione precedente dell'elemento.

Per fare questo possiamo sfruttare i quaternioni, uno strumento matematico potente in quanto capace di esprimere qualsiasi orientazione nello spazio di un corpo rigido.

³¹ La funzione che si occupa del calcolo dell'istante di distanza minima tra due segmenti è riportata nel file linedist.s3d nella cartella include.

(Bowman et al., 2004). In particolare un quaternione è un vettore a quattro dimensioni definito come:

$$q = \left[\vec{v}, w \right] = [(q_x, q_y, q_z), q_w]$$

dove w è un numero reale (scalare) che rappresenta l'entità della rotazione, mentre \vec{v} è un vettore a tre dimensioni che esprime la direzione della rotazione.

Per la precisione una rotazione su un asse unitario \vec{v} (versore) di un angolo θ , si può esprimere con il seguente quaternione unitario di rotazione:

$$q = \left[\sin\left(\frac{\theta}{2}\right) \vec{v}, \cos\left(\frac{\theta}{2}\right) \right]$$

Le rotazioni espresse come quaternioni possono essere combinate tra loro attraverso una moltiplicazione tra quaternioni. Quindi, se per esempio un oggetto all'istante 0 ha un certo orientamento q_0 , poi all'istante 1 si applica una rotazione all'oggetto (rispetto all'orientamento q_0) del quaternione delta_quat , è possibile esprimere l'orientazione complessiva

$$q_1 = q_0 * \text{delta_quat};$$

Notare che il prodotto tra quaternioni non è commutativo.

Quanto esposto può essere utilizzato nella nostra implementazione del quaternione di rotazione. Dell'elemento corrente possediamo l'orientazione corrente espressa in notazione angolo,asse. Come spiegato sopra, dalla notazione angolo asse si può passare facilmente al quaternione che esprime la stessa rotazione. Allo stesso modo la rotazione calcolata rispetto alla posizione corrente, espressa nella notazione angolo asse può essere espressa come quaternione. La moltiplicazione tra il quaternione della posizione corrente e quello che rappresenta la rotazione ci permette di ottenere un quaternione che esprime la rotazione globale dell'elemento. Questa può essere trasformata nuovamente in notazione angolo-asse e applicata all'oggetto.

Impostazione manuale dei valori di posizione, scalatura e rotazione

A seconda dell'oggetto selezionato, apparirà nel pannello di editing un'interfaccia che consente di inserire manualmente dei valori numerici per impostare la posizione, la

scalatura/dimensione e la rotazione dell'oggetto. Una volta inserito il valore e premuto il tasto *transform* verrà inviato il comando `[set_transform,<pos>,<rot>,<scale>]` per il settaggio di tali valori. L'impostazione manuale dei valori è utile nel caso si vogliamo effettuare trasformazioni precise.

3.3.8 Salvataggio dell'IL su server web

Per il salvataggio è sufficiente indicare il nome che si vuole dare al progetto e fare un clic sul bottone *SaveAs*. Questa interazione determinerà l'invio alla pagina web di un comando ['save']. Una volta ricevuto tale comando, la pagina web procede con la generazione di una stringa in formato binario che descrive l'information landscape correntemente visualizzato (metodo *ILWorld::ILDump()*), e invia tale stringa in codifica esadecimale al codice javascript presente nella pagina web insieme ad altre importanti informazioni. Per fare questo viene utilizzato il seguente comando:

```
["save",<hexstring>,<UploadedFile>];
```

dove <hexstring> è la stringa esadecimale che descrive l'IL, <UploadFile> è una stringa che contiene i path delle risorse (immagini, modelli 3D) necessarie.

La codifica esadecimale è stata necessaria perché effettuando alcuni tentativi ci si è accorti che la funzione *DataOut* è in grado di trasferire correttamente solo i caratteri che rientrano nella codifica ASCII (da 0 a 127), mentre nel caso di caratteri con codice maggiore di 127 essa trasferisce il carattere "?". Volendo essere precisi, neanche tutti i caratteri della codifica ASCII vengono trasferiti correttamente: per esempio il codice 0 che corrisponde al carattere di controllo NULL viene anch'esso sostituito dalla *dataOut* con un "?".

È necessario poi inviare al server web la stringa ricevuta. Per fare questo è stato utilizzato AJAX (Asynchronous Javascript and XML), una tecnica che consente lo scambio di dati in background fra pagina web e web browser, e in particolare il metodo di invio POST dato che consente l'invio di un numero maggiore di caratteri (solitamente pari a 8MB, ma dipendente dalla configurazione del web server) rispetto al metodo GET che impone un limite di 256 caratteri per invio. Poiché il file ILXB generato è potenzialmente molto grande, si è pensato di effettuare più richieste POST successive.

Lo scambio dal lato javascript è gestito dalla funzione *Sendviapost(ilxb, numbyte, ilnew, ilold, uploadedfile)*, in cui il parametro *ilxb* contiene la stringa che descrive l'IL (o meglio, una porzione di essa), *numbyte* è il numero di bytes che vogliamo inviare per ogni richiesta post, *ilnew* è il nome con cui vogliamo dare al nuovo IL, *ILold* serve a comunicare al server il nome dell'IL a partire dal quale si è costruito il nostro IL, *UploadedFile* deve contenere l'elenco dei filename delle risorse caricate in fase di creazione. Tale funzione effettua una serie di richieste post inviando per ciascuna una porzione di file *ilxb* più altre informazioni necessarie.

Lato server, è presente uno script php (*ilxsave.php*) che riceve ogni blocco di dati, concatenando le varie porzioni di stringa *ilxb*. Una volta che ha ricevuto tutte le richieste procede nel seguente modo:

- 1) duplica il file zip del vecchio IL creandone un altro avente però il nome richiesto per il salvataggio
- 2) inserisce nella nuova cartella zip le eventuali risorse aggiuntive presenti nel nuovo IL
- 2) rimuove il vecchio file *ilxb* dalla nuova cartella zip
- 3) decodifica la stringa ricevuta e aggiunge il nuovo file *ilxb* nella nuova cartella zip

Per la gestione dei file zip si utilizza la classe *ZipArchive()* resa disponibile dalla libreria *php_zip.dll* supportata nativamente da PHP a partire dalla versione 5.2.2. Nel caso si utilizzi una versione precedente di PHP sarà necessario assicurarsi che tale estensione sia installata e abilitata nel file di configurazione.

Una volta che il file è stato salvato su server web, lo script javascript si occuperà di aggiungerlo tra le opzioni della select box in alto a sinistra, che mostra tutti gli ILs disponibili.

La seguente illustrazione (Figura) riassume l'operazione di salvataggio di un IL.

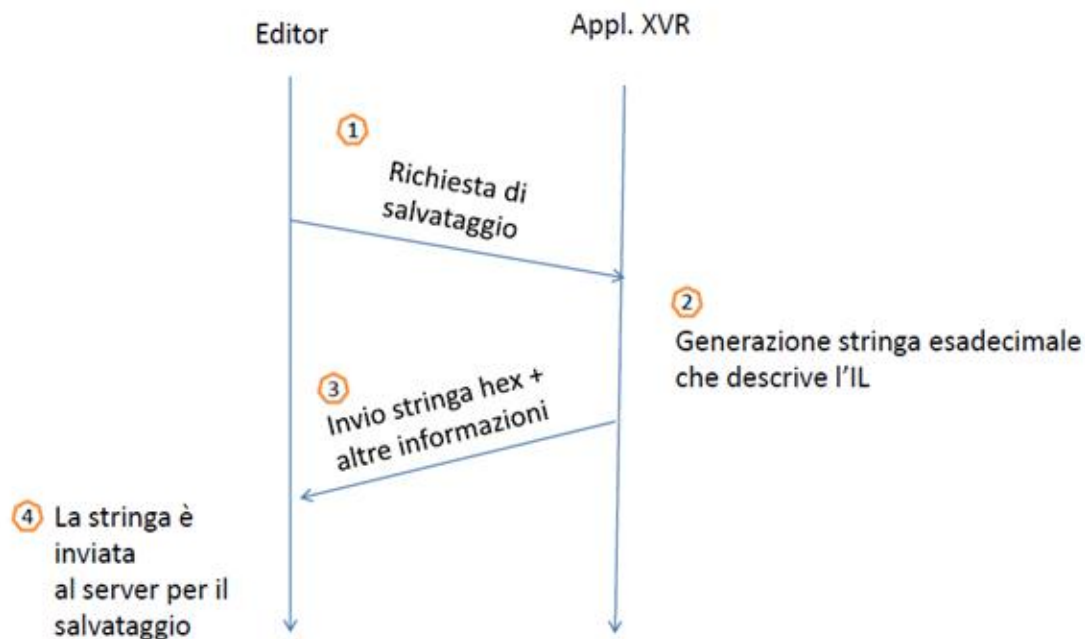


Figura 45 Schematizzazione del meccanismo di salvataggio

3.3.9 Caricamento e cancellazione di un IL da server web

Gli ILs attualmente disponibili sul server sono visualizzati tra le opzioni della select-box presente in alto a sinistra. Le opzioni sono infatti inserite al momento del caricamento della pagina web per mezzo di uno scambio di informazioni asincrono tra web server e pagina web, utilizzando a tal scopo la tecnica AJAX³².

Di default l'IL caricato è quello denominato appunto *default.zip* contenente la scritta "Hello World from ILXVR". L'utente potrà però decidere di caricare uno qualsiasi tra gli IL disponibili, selezionando quello desiderato tramite la select-box e premendo il bottone LoadIL. Tale pressione determinerà l'invio del comando ['loadIL',<ILname>] dove <ILname> indica il path sul server dell'IL che vogliamo caricare.

L'applicazione XVR, una volta ricevuto tale comando, predispone il caricamento asincrono del file tramite la funzione *FileDownload()* e, una volta completato il caricamento, la funzione *DownLoadReady()* si occupa di richiamare la funzione *unpackilxb()* (file *iloader.s3d*) per il caricamento del nuovo IL. Dopodiché l'IL è disponibile sullo schermo e vi si può interagire.

³² Di questo si occupa lo script PHP *LoadFileList.php*.

Al momento del caricamento di un nuovo IL da server web, inoltre, vengono inoltre aggiornate le liste di risorse (immagini e modelli 3d) nelle select box relative attraverso un'opportuna richiesta ajax. La richiesta viene gestita dallo script *LoadImageList.php* (nel caso di immagini) e da *LoadModelList.php* (nel caso di modelli 3D). Lo script *LoadImageList.php* esegue le seguenti azioni:

1) Esplora il contenuto dell'archivio zip relativo all'IL caricato e se trova una immagine (file con estensione jpg o png) lo inserisce nella cartella upload/images.

2) esplora il contenuto della cartella upload/images e invia sotto forma di stringa alla pagina web la lista delle immagini presenti (se è presente un file con lo stesso nome, esso viene sovrascritto).

Lato javascript, a partire dalla stringa ricevuta si effettua un'inizializzazione della select box relativa alle immagini.

Un comportamento simile è quello dello script *LoadModelList.php*.

Selezionando un IL a scelta dalla select-box e premendo *deleteIL* è possibile ottenerne la cancellazione dal web server. Il comando infatti attiva una richiesta ajax al server, dove è presente lo script *deleteIL.php* che si occupa della rimozione del file zip dell'IL. Ad avvenuta cancellazione viene rimossa da javascript la option relativa all'IL appena cancellato dalla lista di quelli disponibili per il caricamento. Nel caso in cui si tenti di cancellare il file su cui si sta lavorando verrà mostrato un opportuno messaggio di errore.

3.3.10 Sviluppo della funzionalità di ricerca full text

Per l'implementazione della funzionalità di ricerca testuale, si è deciso di incorporare un database all'interno dell'applicazione XVR. La scelta è ricaduta su SQLite3³³, una libreria che fornisce un motore per la gestione di database relazionali SQL, e che supporta ricerche testuali full-text. SQLite3 è progettata per essere incorporata direttamente all'interno di un'applicazione (come dll, ovvero Dynamic Link Library) anziché essere installato, come avviene di solito come applicazione server. Nel nostro caso la scelta di utilizzare un database che non ha bisogno di installazione su server, ha il vantaggio di poter essere utilizzato anche in locale senza bisogno della connessione web. Tuttavia,

³³ <http://www.sqlite.org/index.html>

almeno per il momento, risulta compromessa la fruizione via web. Il framework XVR, per motivi di sicurezza non accetta l'incorporamento di librerie dinamiche esterne durante l'esecuzione da web a meno che non siano certificate da chi ha sviluppato il framework, e per ottenere questo certificato c'è tutto un iter da seguire. Tuttavia, se in futuro tale dll otterrà tale certificato, la funzionalità di ricerca testuale, sarà disponibile sia in locale sia su web senza che siano necessarie due soluzioni distinte.

SQLite3 ha inoltre il vantaggio di supportare le ricerche full text per mezzo di tabelle speciali dette FTS tables che generano automaticamente e mantengono aggiornato un indice full text, permettendo all'utente di interrogare in modo efficiente un database alla ricerca di quei record che contengono le parole desiderate, anche in presenza di numerosi documenti. È possibile creare una fts table e popolarla tramite le query:

```
-- Create an FTS table
CREATE VIRTUAL TABLE pages USING fts4(title, body);

-- Insert a row with a specific docid value.
INSERT INTO pages(docid, title, body) VALUES(53, 'Home Page',
'SQLite is a software...');
```

In una tabella fts è possibile effettuare una grande varietà di ricerche: ricerca di una singola parola, utilizzo di operatori booleani, ricerca di frasi, e le cosiddette token-prefix queries (ricerca di tutti i termini che iniziano in un certo modo). Per la sintassi si rimanda alla documentazione ufficiale³⁴. Una funzione speciale denominata offsets può essere utilizzata per ottenere per ciascun record che soddisfa la query una lista di informazioni utili. In particolare, per ogni match rilevato all'interno del record corrente, sono restituiti quattro valori interi:

- il primo valore rappresenta il numero della colonna del record in cui vi è un'occorrenza del termine ricercato (a partire da 0)
- il secondo valore rappresenta il numero del termine trovato tra quelli espressi all'interno dell'espressione di ricerca.
- il terzo valore rappresenta l'offset espresso in bytes del termine trovato all'interno della colonna

³⁴ <http://www.sqlite.org/fts3.html>

- il quarto valore rappresenta la lunghezza espressa in bytes del termine trovato

Per maggiore chiarezza riportiamo un esempio di utilizzo della funzione `offsets` tratta dalla documentazione di `sqlite`, dove sono presenti comunque anche altri esempi per chi fosse interessato.

```
CREATE VIRTUAL TABLE mail USING fts3(subject, body);
INSERT INTO mail VALUES('hello world', 'This message is a hello
world message. ');
INSERT INTO mail VALUES('urgent: serious', 'This mail is seen as
a more serious mail');

-- The following query returns a single row (as it matches only the first
-- entry in table "mail". The text returned by the offsets function is
-- "0 0 6 5 1 0 24 5".
--
-- The first set of four integers in the result indicate that column 0
-- contains an instance of term 0 ("world") at byte offset 6. The term instance
-- is 5 bytes in size. The second set of four integers shows that column 1
-- of the matched row contains an instance of term 0 ("world") at byte offset
-- 24. Again, the term instance is 5 bytes in size.
SELECT offsets(mail) FROM mail WHERE mail MATCH 'world';
```

Prima di procedere con la spiegazione di come è stata implementata la funzionalità di ricerca su `IL`, facciamo una breve digressione su come `XVR` gestisce l'incorporamento di una `dll` esterna. `XVR` importa una libreria esterna come oggetto utilizzando il comando:

```
library = CVmExternDLL("myLib.dll ");
```

Per importare le funzioni messe a disposizione della `dll` si utilizza il metodo `__AddFunction()`. Tale metodo vuole come parametri il tipo di valore di ritorno della funzione importata, il nome della funzione importata, e il tipo dei parametri:

```
library.__AddFunction(C_FLOAT, "myFunction", C_PFLOAT, C_INT);
```

Tali funzioni saranno poi accessibili come metodi dell'oggetto `CVmExternDLL`:

```
var a = [1.0, 2.0, 3.0];
var b;
b = library.myFunction(a, 3);
```

Per maggiori dettagli si veda la documentazione XVR disponibile sul web³⁵.

Il codice relativo all'incorporamento e gestione di un database sqlite3 all'interno dell'applicazione XVR, è contenuto nella `SQLite3Db` che rappresenta il database creato e mette a disposizione alcuni metodi per l'esecuzione di query SQL, la restituzione del risultato di una ricerca, il controllo di validità del database, la chiusura del database.

```
class SQLite3Db
{
    var db; // db handle

    exec(name); // executes any SQL statement with no results
    isValid(); // check if valid
    close(); // closes db
    prepare(sql); // returns SQLite3Stmt for sql statement
};
```

È presente inoltre una seconda classe che rappresenta il risultato della query sql.

```
class SQLite3Stmt
{
    var st; // handle

    close(); // close
    rewind(); // reset the statement
    step(); // next step
    getColumnCount(); // number of columns

    bindArray(a); // binds a full array to parameters
    bind(i,value); // bind by index or name the given value
    getParamCount(); // number of parameters

    getText(icol); // return text column
    getInt(icol); // return int column
    getFloat(icol); // return float
    getName(icol); // return column name
    getNames(); // returns column names
    getArray(); // returns as array
    get(i); // returns given value
};
```

Essa mette a disposizione vari metodi per l'estrazione delle singole informazioni dal risultato complessivo. In particolare, noi si è frequentemente utilizzato il metodo `step()`

³⁵ http://wiki.vrmedia.it/index.php?title=XVR_Documentation

che consente di scorrere i vari record che costituiscono il risultato, il metodo *getArray()* che restituisce il record corrente come array di valori, e il metodo *get()* che restituisce il valore presente all'i-esima posizione del record corrente.

A partire da queste premesse si è implementata la ricerca testuale come spiegato di seguito. Buona parte del codice relativo alla gestione della ricerca testuale è stato raggruppato nel file *ildatabase.s3d*.

Al momento dell'avvio dell'applicazione, nella funzione *OnInit()* viene invocata la funzione *CreateILdb(world,&db)* per la creazione del database. Il primo parametro passato costituisce la radice dell'IL mentre il secondo parametro, passato per indirizzo, è una variabile globale che conterrà alla fine dell'esecuzione della funzione un riferimento al database appena creato. La funzione *CreateILdb(world,&db)* si occupa di creare un database e due tabelle: la tabella *item* che contiene, per ogni oggetto un record contenente il path dell'elemento e tutto il testo che lo costituisce (il caption nel caso di immagini); la seconda tabella *item2* in cui per ogni elemento testuale sono memorizzati più record uno per ogni riga di testo, contenente il path dell'oggetto, il numero di riga, la porzione testuale contenuta in quella riga. Tali tabelle sono quindi popolate ciascuna con i record previsti (Tabella 1, Tabella 2).

path	testo
-------------	--------------

Tabella 1 Struttura del record della tabella *item*

path	riga	testo
-------------	-------------	--------------

Tabella 2. Struttura del record della tabella *item2*

Si è pensato di creare due tabelle perché, come si è detto, abbiamo voluto creare due modalità di ricerca, una che a partire da una query testuale dell'utente evidenzia globalmente tutti gli oggetti (*iltext*, *ilimage*) che soddisfano la ricerca, e un'altra che evidenzia solamente le singole parole che soddisfano la ricerca. Poiché il sistema di rendering tratta il testo presente in un *ILTEXT* riga per riga era infatti utile memorizzare le singole righe del testo, anziché tutto il blocco di testo, in modo da ricevere i risultati della ricerca relativamente alle singole righe e non all'intero elemento.

Quando l'utente inserisce nella casella di testo un' espressione da ricercare e preme il tasto cerca, il seguente comando viene inviato all'applicazione XVR

```
[ 'search', <text> ]
```

dove il parametro <text> rappresenta l'espressione testuale ricercata. L'applicazione XVR una volta ricevuto il comando, effettua una chiamata ad una funzione di ricerca. Se è stata richiesta la ricerca con visualizzazione single-words, allora viene chiamata la funzione *Search_text2(db,text)* altrimenti viene invocata la funzione *Search_text(db,text)*.

La funzione *Search_text* è più semplice: essa effettua una semplice query allo scopo di ricercare tutti i documenti che soddisfano la richiesta:

```
SELECT * FROM item WHERE text MATCH 'expression'
```

Una volta ottenuto il risultato si scorrono tutti i record presenti, e per ciascuno di essi si imposta il colore per la proprietà *search_color* dell'elemento a cui il record si riferisce (determinato a partire dal path memorizzato nel record). In fase di rendering infatti, se un oggetto possiede la proprietà *search_color* diverso da *void*, allora significa che l'oggetto è un risultato di una ricerca e va perciò evidenziato con colore *search_color*. Le immagini sono evidenziate tramite un riquadro colorato, mentre il testo è evidenziato disegnandolo con colore *search_color* anziché il normale colore *color*.

Più complicato è il comportamento per quanto riguarda la funzione *Search_text2()*. Per prima cosa essa effettua la seguente query:

```
SELECT path, row, offsets(item2) FROM item2 WHERE text MATCH  
'expression' ORDER BY path,row;
```

allo scopo di ottenere per ogni riga di testo che soddisfa la richiesta, una stringa di valori che mi indicano per ogni match rilevato alcune informazioni importanti come descritto sopra in questo paragrafo.

Allo scopo di poter effettuare la visualizzazione, tali informazioni devono poter essere accedute in modo efficiente dal sistema di rendering per la visualizzazione del risultato della ricerca. Un modo poteva essere quello di creare una struttura dati ad accesso rapido come ad esempio le tabelle hash. Si è ritenuto però che una soluzione più semplice fosse

quella di memorizzare tali informazioni direttamente come proprietà di ciascun oggetto risultato della ricerca.

All'interno del file *iltext.s3d*, quindi, per prima cosa sono state create due classi aggiuntive che sono:

```
class Block
{
    var inizio;
    var acceso;
};

class HighLine
{
    var linea;           //number of text line (start from 0)
    var blocks;         //array of block object
};
```

La classe *HighLine* rappresenta una linea di testo risultato della ricerca: la variabile d'istanza *linea* rappresenta il numero identificativo di linea, mentre la variabile *blocks* è un array di oggetti della classe *Block* ciascuno rappresentante una porzione di testo all'interno della linea (che va dalla posizione *inizio* fino all'inizio del blocco successivo) e come questa porzione debba essere disegnato (se *acceso* vale true significa che il testo dovrà essere disegnato evidenziato di colore *search_color*, altrimenti sarà testo normale visualizzato con colore *color*). Si è inoltre stabilito che il primo blocco di una riga debba essere sempre (0, *acceso*) e l'ultimo (*lunghezza(linea)*, false). Nel caso in cui si abbia la riga "Ciao Mondo" con Mo da evidenziare avremmo quindi concettualmente i seguenti blocchi: (0, false) (5, true) (7, false) (10, false).

All'interno della classe *ILText* invece è stata aggiunta la proprietà *highLines*, un array di oggetti della classe *HighLine*, contenente un elemento per ogni linea di testo risultato della ricerca con le relative informazioni per la visualizzazione.

Per quanto riguarda la visualizzazione dei risultati della ricerca abbiamo dovuto intervenire sul meccanismo di rendering eseguito nella funzione *Draw(ctx)* della classe *ILText*, in modo che se è stata effettuata una ricerca *singlewords*, visualizzi le parole richieste in un colore diverso dalle altre.

È importante osservare che il database viene mantenuto costantemente aggiornato in seguito ad aggiunte, modifiche e cancellazioni di elementi nel paesaggio.

4 Usabilità del visualizzatore XVR di paesaggi di Informazione

Ai fini di una maggiore diffusione dei Paesaggi di Informazione, lo sviluppo di strumenti di editing accessibili via web non basta. Occorre anche che il pubblico accetti questa nuova modalità di divulgazione e fruizione delle informazioni. Al momento, gli Information Landscapes sono rimasti confinati all'interno del mondo della ricerca, dove si è puntato più al loro funzionamento dal punto di vista tecnico, senza prestare particolare attenzione alla loro usabilità, requisito fondamentale perché questi sistemi siano accettati presso il grande pubblico. Questa tesi si propone di fare un piccolo passo nella direzione di una maggiore usabilità dell'attuale visualizzatore per ILs sviluppato presso il Laboratorio PERCRO della Scuola Superiore S. Anna, sottoponendolo al giudizio di un campione di utenti per mezzo di un esperimento di usabilità. Tale esperimento si pone l'obiettivo di identificare quelle che sono le maggiori problematiche di usabilità, ipotizzarne delle soluzioni, e in alcuni casi implementarle.

La prima parte del capitolo fornirà i fondamenti teorici che sono stati necessari per la progettazione del test di usabilità. In particolare sarà fornita una definizione del concetto di usabilità, sarà descritto in che cosa consista una valutazione di usabilità, analizzando i principali metodi, approcci e metriche di valutazione. Infine si passerà alla descrizione della fase di preparazione dell'esperimento motivando le scelte fatte, e riportando i risultati.

4.1 Usabilità: aspetti teorici

4.1.1 Definizione di usabilità

In letteratura esistono varie definizioni di usabilità. La definizione che forse esprime in modo più preciso e completo il concetto è quella riconosciuta all'interno dello standard ISO 9241, standard in cui sono definiti i "requisiti ergonomici per il lavoro di ufficio con terminali". L'usabilità è qui espressa come la *misura* in cui un *prodotto* può essere usato da specifici *utenti* per raggiungere determinati *obiettivi* con *efficienza*, *efficacia* e *soddisfazione*, dove per efficienza si intendono le risorse spese in relazione all'accuratezza e completezza degli obiettivi raggiunti (es: il tempo richiesto), per efficacia si intende l'accuratezza e la completezza con cui gli utenti possono raggiungere i loro obiettivi e infine, per soddisfazione si intende il comfort e l'accettabilità del sistema per gli utenti e le altre persone influenzate dal loro uso.

Molti sono i motivi per cui l'usabilità di un prodotto è importante. Più un prodotto è usabile più aumenterà, per esempio, l'efficienza degli utenti e la produttività nelle aziende e organizzazioni (nel caso di un prodotto ad uso aziendale), diminuiranno gli errori e quindi la sicurezza, si ridurrà il bisogno di training e di supporto agli utenti (a cui spesso sono associati dei costi), si determinerà una maggiore accettazione nell'uso di quella applicazione informatica, e nel caso di un prodotto commerciale aumenteranno le vendite.

4.1.2 Valutazione di usabilità

Per valutazione di usabilità si intende il giudizio di un artefatto dal punto di vista dell'usabilità. Nel campo della Human Computer Interaction³⁶ (HCI) l'artefatto consiste in un prodotto informatico interattivo, dotato cioè di un'interfaccia verso l'utente.

Lo scopo principale della valutazione è l'identificazione di problemi di usabilità, da cui partire per effettuare modifiche e miglioramenti al progetto. Progettazione e valutazione dovrebbero rientrare quindi in un processo iterativo: la progettazione è seguita dalla valutazione, la quale porta ad una ri-progettazione, che poi può essere valutata, e così via. L'iterazione finisce quando l'interfaccia utente è abbastanza buona secondo le metriche

³⁶ Secondo una definizione contenuta in uno studio specialistico dal titolo *ACM SIGCHI Curricula for Human Compute- Interaction*, la HCI "è una disciplina che ha a che fare con la progettazione, la valutazione e l'implementazione di sistemi informatici interattivi per uso umano e con lo studio dei maggiori fenomeni a essi connessi". La HCI è un'area di ricerche interdisciplinari che coinvolge diverse discipline: l'informatica, la psicologia, la sociologia, l'antropologia, la progettazione industriale.

che abbiamo stabilito (o, più di frequente nella realtà, quando il budget finisce oppure arriva la scadenza). Scopi secondari possono essere lo sviluppo di linee guida generali per il design, o lo sviluppo di “performance models”.

Il processo di valutazione dell'usabilità necessita prima di tutto di essere attentamente progettato: molti sono i **metodi** (o tecniche) di valutazione esistenti e la scelta di quale metodo utilizzare dipende da vari fattori (obiettivi, budget, costi ecc). Bisogna inoltre precisare che spesso metodi di valutazione differenti vengono combinati in una particolare sequenza (**evaluation approach**) in modo da ottenere una valutazione più completa.

La persona che si occupa della progettazione e dell'esecuzione del test (così come della valutazione finale dei risultati) prende il nome di **valutatore**.

Una volta terminata la fase di progettazione, il processo di valutazione ha inizio. A seconda della metodologia scelta, può essere necessaria la partecipazione di un certo numero di **utenti**, cioè di persone esterne opportunamente selezionate e chiamate ad utilizzare l'interfaccia, svolgere determinati compiti (tasks) o rispondere a domande. I dati raccolti durante il test, sono analizzati dal valutatore.

Gli aspetti del progetto che vengono considerati per la valutazione sono detti **attributi di usabilità**. Il risultato di una valutazione di usabilità consiste in una serie di possibili **misure**³⁷ dell'usabilità dell'interfaccia, che possiamo raggruppare in tre insiemi distinti: quelle che misurano le prestazioni del sistema (**system performance**), quelle che misurano le prestazioni dell'utente nello svolgere i tasks (**user task performance**), e altre che rientrano nella sfera più personale dell'utente (**user preferences**).

4.1.3 Metodi e approcci per la valutazione dell'usabilità

In generale esistono molti metodi di valutazione dell'usabilità. Sviluppati inizialmente per l'usabilità delle interfacce grafiche 2D, essi sono stati successivamente estesi per supportare la valutazione delle interfacce 3D³⁸. Una utile classificazione dei metodi di valutazione dell'usabilità è quella fornita da Bowman et al. (2005). I metodi sono

³⁷ Alcuni autori fanno coincidere attributi di usabilità e metriche o misure. Noi preferiamo mantenere distinti gli aspetti oggetto di valutazione e il modo di misurare tali aspetti.

³⁸ La valutazione delle interfacce 3D differisce da quella delle interfacce 2D. Per un approfondimento si veda Bowman et al. 2005.

classificati in uno spazio secondo tre caratteristiche: *coinvolgimento degli utenti*, *contesto di valutazione* e *tipo di risultati prodotti* (Figura 41).

La prima caratteristica distingue tra metodi che richiedono la partecipazione di utenti e quelli invece che non richiedono il coinvolgimento degli utenti, bensì quello di uno o più esperti di usabilità. La seconda caratteristica descrive il tipo di contesto in cui ha luogo la valutazione. Ci sono metodi che possono essere applicati in un contesto generico e altri che sono applicati nell'ambito della valutazione di un'applicazione specifica. In generale, i risultati di valutazioni condotte in un contesto generico sono più estendibili di quelle ottenute nell'ambito della valutazione di una applicazione specifica. La terza caratteristica specifica se un metodo produce principalmente risultati di tipo qualitativo o quantitativo.

Sono state scelte queste tre caratteristiche perché ritenute le più significative per i valutatori che intendono progettare una valutazione di usabilità, i quali vorranno conoscere quale sarà il costo della valutazione (direttamente dipendente dalla presenza degli utenti), quali saranno i risultati della valutazione, e il grado di estendibilità dei risultati (dipendente dal contesto di valutazione).

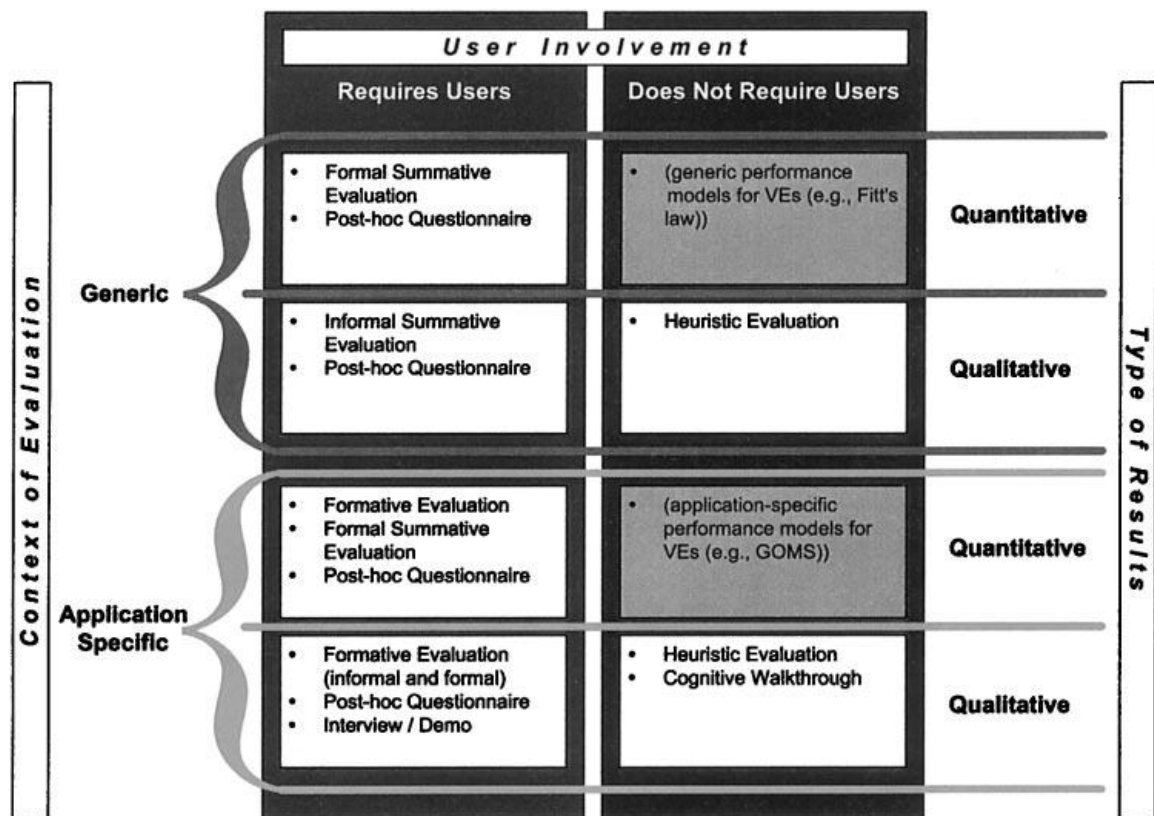


Figura 41 Classificazione dei metodi di valutazione dell'usabilità per le interfacce 3D

COGNITIVE WALKTHROUGH (Polson et al., 1992)

È un metodo di ispezione dell'usabilità condotta da uno o più esperti di usabilità, che mira a prevedere i problemi che gli utenti possono incontrare utilizzando l'interfaccia per la prima volta, senza addestramento.

Per eseguire tale tipo di test è necessario avere un prototipo dell'interfaccia, conoscere i tasks che possono essere eseguiti con quella interfaccia e i vari passi necessari a portarli a termine, le caratteristiche dell'utente potenziale e il contesto d'uso. I valutatori eseguono la sequenza di azioni per ogni task, e nel fare questo per ogni azione cercano di rispondere a varie domande come:

- L'utente è in grado di formulare gli obiettivi corretti?
- L'azione corretta è sufficientemente chiara e visibile all'utente?
- Dopo aver eseguito l'azione corretta, il feedback ottenuto è in grado di indicare all'utente che essa era in effetti la cosa giusta da fare?

HEURISTIC EVALUATION (Nielsen and Molich, 1992)

È un metodo in cui diversi esperti di usabilità valutano separatamente un'interfaccia applicando un insieme di euristiche o linee guida che sono abbastanza generali da valere per qualsiasi UI o sono specifiche per le interfacce 3D³⁹. I risultati da parte degli esperti sono poi combinati e i problemi ordinati secondo una lista di priorità.

FORMATIVE EVALUATION (Hix and Hartson 1993)

È un metodo di valutazione empirico, applicato durante la fase di sviluppo o miglioramento dell'applicazione in cui un certo numero di utenti rappresentativi sono posti di fronte all'applicazione e richiesto loro di svolgere determinati task. In questo modo è possibile identificare problemi di usabilità o comunque di valutare o misurare la capacità di un certo design a supportare l'utente nell'esecuzione dei task e nell'apprendimento, la user satisfaction ecc..

Le valutazioni possono essere *informali* o *formali*. Se informali – che significa che l'utente è osservato durante il libero uso dell'interfaccia, senza il vincolo di tasks predefiniti - puntano principalmente a ottenere risultati qualitativi come commenti

³⁹ Mentre esistono euristiche generali, come le euristiche di Nielsen, mancano invece linee guida standard o euristiche precise per la progettazione e la valutazione delle interfacce 3D.

dell'utente o reazioni generali, situazioni critiche; se formali, si pongono l'obiettivo di produrre risultati sia qualitativi che quantitativi (tempo di esecuzione dei task, errori, ecc).

I passi da compiere per svolgere una valutazione formativa sono:

- Preparare l'esperimento
- dirigere la sessione di valutazione
- collezionare i dati
- analizzare i dati
- individuare i problemi
- riprogettare e implementare l'interfaccia

La formative evaluation deve essere svolta più volte durante il processo: almeno 3 cicli di valutazione seguita da una riprogettazione sono necessari (Hix e Hartson, 1993)

SUMMATIVE EVALUATION (Hix and Hartson 1993, Scriven 1967)

È un metodo di valutazione empirico, applicato dopo che l'interfaccia (o parte di essa) è completa. Come nella formative evaluation, utenti rappresentativi sono chiamati a svolgere certi task e i valutatori collezionano dati sia quantitativi che qualitativi. Possono essere informali o formali. Spesso usati per confrontare due o più configurazioni di UI design e UI techniques.

QUESTIONARI (Hix and Hartson 1993)

È un insieme scritto di domande usate per ottenere informazioni dagli utenti prima o dopo la sessione di valutazione. Utili soprattutto per ottenere informazioni demografiche e opinioni soggettive. Molto importanti nell'ambito delle interfacce 3D per ottenere informazioni su fenomeni soggettivi come presenza o cybersickness.

INTERVISTE (Hix and Hartson 1993)

È una tecnica per raccogliere informazioni sugli utenti dialogando direttamente con loro. Un'intervista permette di ottenere un numero maggiore di informazioni rispetto ai questionari.

Possono essere di due tipi: *strutturate*, se prevedono un insieme predefinito di domande e risposte; *aperte* che permettono maggiore libertà sia all'intervistatore che all'intervistato.

Si tratta di un metodo non studiato esplicitamente nel contesto della valutazione 3D ma spesso utilizzato alla fine delle formative e summative evaluations.

Al fine di ottenere dei risultati più completi rispetto a quelli che l'applicazione di un singolo metodo di valutazione potrebbe portare, nell'ambito della ricerca sulle 3D UI sono stati studiati dei veri e propri approcci che consistono nella combinazione di più metodi. Alcuni approcci sono più generali, nel senso che puntano alla valutazione di una tecnica di interazione al di fuori del contesto di un'applicazione (per esempio in un contesto generale); altri invece sono specificatamente diretti a testare l'usabilità di applicazioni specifiche. Al primo tipo appartiene il cosiddetto "Testbed Approach" (Bowman and Hodge, 1999), mentre al secondo tipo appartiene un altro approccio denominato "Sequential Approach" (Gabbard, Hix, Swan, 1999).

4.1.4 Attributi di usabilità

Hix et al. (1993) hanno individuato una lista di attributi di usabilità comuni, cioè di aspetti del prodotto che possono essere sottoposti a valutazione. Essi sono:

- *initial performance*: le prestazioni dell'utente durante il primo uso dell'interfaccia
- *Long term performance* : le prestazioni dell'utente dopo un costante uso dell'interfaccia
- *Learnability*: quanto facilmente e velocemente l'utente impara ad usare il sistema
- *Retainability*: quanto l'utente trattiene ciò che ha imparato dopo un certo periodo di tempo
- *Advanced feature usage*: misura l'usabilità di funzioni complesse dell'interfaccia
- *First impression*: cattura l'opinione iniziale del sistema da parte dell'utente
- *Long term user satisfaction*: si riferisce all'opinione sul sistema dopo che si è usato per un lungo periodo di tempo

Nelle interfacce 3D in particolare, possiamo aggiungere altri aspetti importanti che hanno invece poco senso nella valutazione delle interfacce 2D tradizionali come la *presenza* e lo *user comfort*.

Anche le prestazioni del sistema possono essere valutate nella misura in cui influenzano l'esperienza dell'utente.

4.1.5 Metriche di valutazione

Le metriche di valutazione di un'interfaccia utente possono essere raggruppate in tre insiemi: quelle che misurano le prestazioni del sistema (*system performance*), quelle che misurano le prestazioni dell'utente nello svolgere i tasks (*user task performance*), e altre che rientrano nella sfera più personale dell'utente (*user preferences*).

System Performance Metrics – si riferiscono alle misure delle prestazioni del computer o del sistema grafico e sono per es. il frame-rate medio, latenza media, ritardo di rete, distorsione ottica.

Task Performance Metrics – si riferiscono alle misure della qualità nello svolgimento di specifici task in un'applicazione 3D da parte di un utente. Comprendono tra le altre cose il tempo impiegato per navigare in un preciso punto, l'accuratezza nel posizionamento di un oggetto, il numero di errori che l'utente compie nel selezionare un oggetto da un insieme. Tipicamente *velocità* e *accuratezza* sono le più importanti task performance metrics. È importante però notare che tra velocità ed accuratezza vi è una relazione implicita: l'utente può andare più veloce ma essere meno accurato, oppure l'utente può essere più accurato ma decrementando la velocità. Per gestire questo trade-off esistono diversi modi: quello di richiedere esplicitamente all'utente quale comportamento tenere; oppure quello di richiedere all'utente di svolgere il task la prima volta il più veloce che può, la seconda volta nel modo più accurato possibile, la terza volta cercando di bilanciare velocità e accuratezza.

User Preference Metrics – mirano a fornire una misurazione della percezione soggettiva dell'interfaccia da parte dell'utente, relativamente ad aspetti come *facilità di uso*, *facilità di apprendimento*, *soddisfazione*.

Queste percezioni sono spesso misurate attraverso questionari o interviste e possono essere sia qualitative che quantitative. Sono metriche importanti in quanto determinano il grado di accettazione dell'applicazione da parte dell'utente, che se basso non verrà utilizzata.

La presenza – “the feeling of being there” – può essere misurata in vari modi: chiedendo agli utenti di quantificare la loro sensazione di presenza in una scala da 1 a 100; usando questionari per ricevere giudizi su vari aspetti della presenza; usando misure fisiologiche, test di memoria, misurando lo user performance nello svolgimento di un task per il quale è necessaria la presenza.

L'attributo dello user comfort include vari aspetti differenti. Il più studiato è il *simulator sickness*, quell'insieme di sensazioni sgradevoli che si producono durante e dopo l'utilizzo di un sistema di realtà virtuale, spesso dovuta ad una mancanza di sincronizzazione tra le varie informazioni sensoriali. Il simulator sickness può essere valutato oggettivamente, misurando ad esempio la capacità degli utenti di svolgere certe azioni nell'ambiente reale dopo essere stati esposti all'ambiente virtuale, oppure soggettivamente attraverso rating-scales o questionari.

4.2 Preparazione dell'esperimento di usabilità sugli ILs

4.2.1 Esperimento preliminare: test di comprensione

Un esperimento preliminare che ho avuto modo di supervisionare nel corso della tesi è quello eseguito da una studentessa di Informatica Umanistica Marina Belli. Tale esperimento si è occupato per lo più di valutare la comprensione testuale all'interno di un Information landscape rispetto ad un mezzo digitale più tradizionale come un ipertesto su browser web. A tal fine, il primo passo è consistito nella scelta del testo a partire dal quale sviluppare il paesaggio e l'ipertesto. La scelta è ricaduta su un testo ricco di collegamenti interni, il racconto "Storia dei Tre Pomi" tratto da "Le Mille e Una Notte". La storia è stata suddivisa in dieci sequenze, di cui le prime quattro si riferiscono alla storia dei tre pomi, e le altre sei alla storia della dama trucidata. Sono state inoltre inserite anche sequenze di approfondimento esterne alla storia, per la comprensione di alcuni termini più difficili presenti nella storia.

Di ognuna di queste sequenze è stata creata una pagina web. Le parole chiave sono state create di colori diversi e costituiscono dei link con cui l'utente può muoversi ad altre sequenze correlate (della storia o di approfondimento). Inoltre, tra le varie sequenze consecutive sono stati inseriti dei collegamenti ("avanti" e "indietro") che consentono all'utente di muoversi linearmente nella storia.

Della storia è stato inoltre costruito un Paesaggio di Informazione (Figura 42). Le prime quattro sequenze (relative alla storia dei tre pomi) sono state disposte a formare una "stanza" a forma di quadrato, le altre (relative alla storia della dama trucidata) sono state disposte a esagono; infine le sequenze di approfondimento sono state raggruppate in un'altra area disposte a semicerchio. Analogamente alle pagine web, ogni sequenza presenta parole colorate che costituiscono i link che collegano tra loro le sequenze (della storia o di approfondimento). Diversamente dall'ipertesto, l'utente ha la possibilità di esplorare il paesaggio in maniera libera, a prescindere dai link.

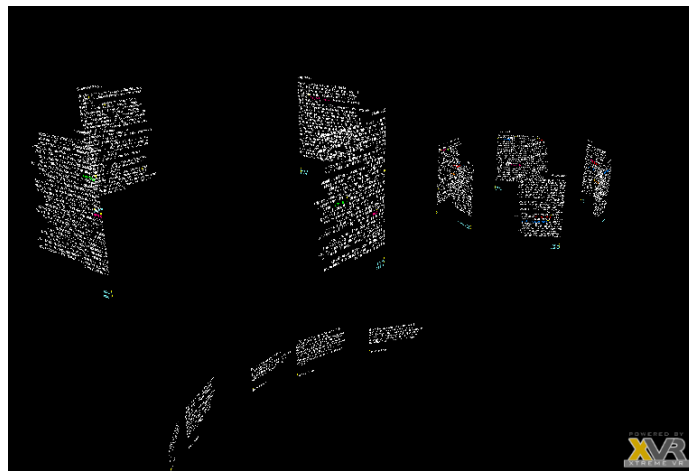


Figura 42 Il paesaggio di informazione "Storia dei tre pomi"

I due sistemi (ipertesto e IL) sono stati sottoposti a un campione di 10 persone (5 per modalità) per un test. Gli utenti sono stati lasciati liberi di leggere il testo per un massimo di 20 minuti. A lettura completata è stato chiesto loro di rispondere ad un questionario di comprensione del testo.

Dall'analisi dei risultati è emerso che coloro che hanno letto il testo attraverso le pagine web hanno avuto una maggiore comprensione rispetto a coloro che hanno fruito del testo attraverso il paesaggio di informazione. Questo è dovuto ad un diverso approccio adottato dagli utenti nella lettura: attraverso le pagine web gli utenti hanno per lo più adottato un approccio di lettura sequenziale, mentre coloro che hanno fruito il testo dal paesaggio di informazione hanno avuto un comportamento più esplorativo, meno orientato alla lettura. Tale approccio ha permesso però agli utenti di cogliere delle informazioni (per esempio, gli approfondimenti) che gli utenti non hanno colto nella modalità ipertestuale. Questo si nota dal maggior numero di riposte esatte date alle domande riguardanti i link di approfondimento rispetto agli utenti che hanno letto il testo sulle pagine web.

Da quanto emerso sembrerebbe poter affermare che gli IL non possono sostituire i mezzi più tradizionali nella comprensione profonda di testi (soprattutto se complessi). Possono però essere affiancati ai mezzi tradizionali per comprendere maggiormente le connessioni.

Questo test preliminare - anche se svolto con il vecchio viewer stand-alone - ha permesso anche di avere un primo feedback sui problemi di usabilità presenti nell'utilizzo degli IL, e hanno reso evidente la necessità di uno studio mirato che si focalizzasse sullo studio dell'usabilità degli IL.

4.2.2 Generalità, obiettivi

Il test si pone l'obiettivo di valutare l'efficacia, l'efficienza e la soddisfazione dell'utente nell'utilizzare l'attuale sistema di visualizzazione e interazione con gli ILs.

Dato che i Paesaggi di Informazione hanno la possibilità di essere utilizzati sia in ambiente desktop (interagendo con il mouse), sia in ambiente immersivo (interagendo con il joystick), è interessante progettare un test che permetta di ottenere misure di entrambe le modalità a scopo di confronto.

L'usabilità nella fruizione di un paesaggio di informazione è dipesa sia dalle funzionalità che il software di visualizzazione mette a disposizione, sia dal modo in cui sono stati strutturati i contenuti da parte del designer. Particolare attenzione dovrà quindi essere dedicata nel classificare i problemi di usabilità individuati come problemi del sistema o come problemi dell'ambiente. I risultati potranno essere utilizzati per ri-progettare eventuali aree problematiche o individuare linee guida riguardo alla costruzione dei paesaggi di informazione che siano usabili.

4.2.3 Metodo di valutazione

Come metodo di valutazione si è deciso di utilizzare la formative evaluation. Tale metodo infatti permette di ottenere una valutazione dell'usabilità e di raccogliere opinioni direttamente dagli utenti, strategia che non è mai stata adottata nel contesto dei Paesaggi di Informazione.

Il primo passo nella progettazione del test è consistito nell'individuazione degli attributi di usabilità cioè degli aspetti da valutare.

Per il nostro test, si è deciso di considerare i seguenti aspetti:

- la *facilità di apprendimento* della tecnica di interazione,
- la *prima impressione*,
- *performance nell'utilizzo dell'interfaccia* di interazione (nell'esecuzione di semplici task di navigazione),
- la *facilità di orientamento* (wayfinding) all'interno dei paesaggi di informazione,
- la *soddisfazione dell'utente* rispetto al sistema.

Di tali aspetti vogliamo ottenere misure sia nel caso di fruizione in ambiente desktop sia nel caso di fruizione in ambiente immersivo in modo da poterli confrontare.

Per ottenere delle misure di tali attributi l'esperimento è stato strutturato in una sessione di training iniziale seguita da alcuni task di navigazione in un ambiente semplice. Seguono alcuni task e di orientamento in un ambiente più complesso e un questionario finale. Nei paragrafi seguenti è riportata una descrizione di ciascuna fase dell'esperimento.

4.2.3.1 Introduzione dell'utente all'esperimento e consenso informato

Un aspetto importante di qualsiasi test è l'introduzione dell'utente all'esperimento. All'utente viene comunicato lo scopo dell'esperimento a cui parteciperà, la sua strutturazione e la sua durata. Il discorso che è stato utilizzato all'inizio del test è riportato in forma scritta in appendice B. Ancora più importante è ottenere il preventivo consenso dell'utente alla partecipazione. Per questo è stato preparato un modulo per il consenso informato, da sottoporre alla lettura dell'utente (appendice C) e di cui è stata richiesta la firma prima di poter iniziare l'esperimento.

4.2.3.2 Sessione di training

L'obiettivo della sessione iniziale di training è quello di permettere all'utente di familiarizzare con la tecnica di interazione che dovrà utilizzare durante il test ma anche permettere una valutazione della *facilità di apprendimento* (learnability) e la prima impressione sul sistema.

Al fine di focalizzare l'attenzione dell'utente sull'apprendimento dell'interazione senza disperderla in altro, la miglior cosa è quella di effettuare questa prima fase su un IL semplice costruito ad hoc e formato solo dagli elementi indispensabili (Figura 43).

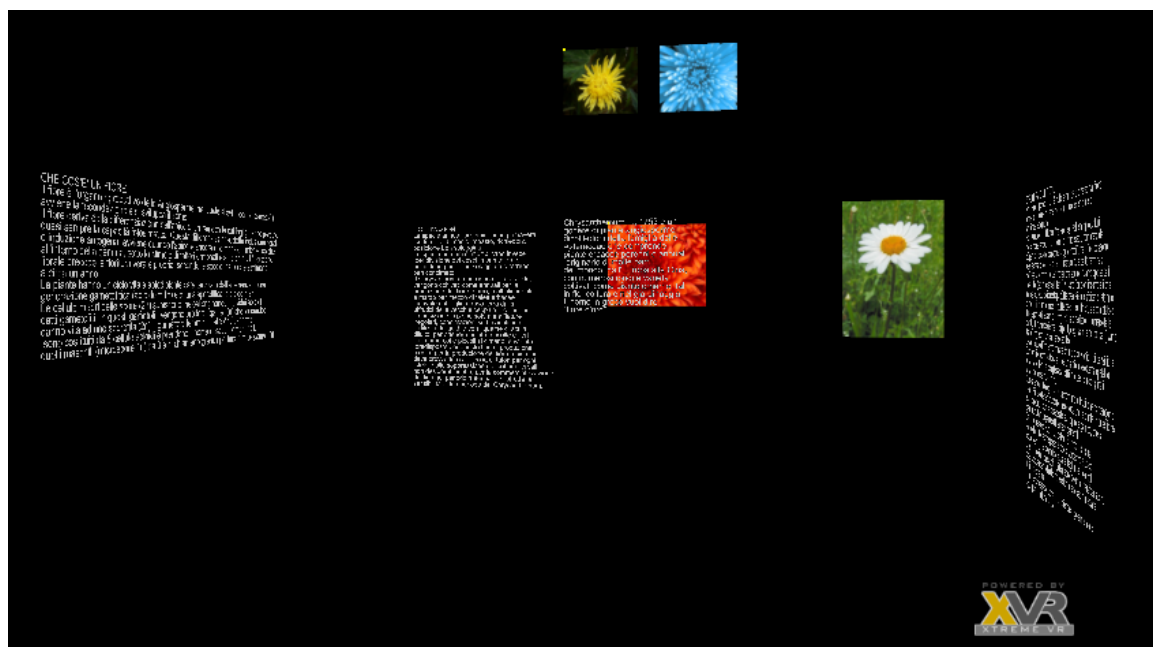


Figura 43 Ambiente usato per il training e per il test di navigazione

Le operazioni che l'utente deve imparare a compiere sono:

- Andare avanti, indietro
- spostarsi a destra, sinistra, in alto e in basso
- Ruotare la camera a destra e a sinistra
- Selezionare un elemento
- raggiungere automaticamente l'elemento selezionato
- Seguire un percorso tramite link

All'utente viene mostrato come si effettua un'azione e viene chiesto di ripeterla. Il valutatore prenderà nota del numero di tentativi effettuati, e di altre eventuali note come richieste di aiuto o osservazioni. Questa procedura avverrà per ogni singola azione. Per prendere nota dei dati il valutatore utilizzerà la seguente tabella (Tabella 3):

Azioni	Numero tentativi	Note
Avanti		
Indietro		
Ruota sx		
Ruota dx		
Destra		
Sinistra		
alto		
basso		
Navigazione automatica		
Segui link		

Tabella 3 Tabella per la registrazione dati durante il training

Una volta terminata questa prima parte, l'utente verrà lasciato libero di impraticarsi con la tecnica di interazione fino a che non dichiara di sentirsi sicuro. In questa fase l'utente potrà richiedere aiuto al valutatore. Il tempo dedicato dall'utente all'addestramento, ed eventuali note (richieste di aiuto, osservazioni dell'utente) verranno riportati nella seguente tabella (Tabella 4):

Tempo di addestramento	Note

Tabella 4 Tabella per la registrazione dei tempi di addestramento

La difficoltà e il comfort *percepiti* dall'utente nell'apprendimento della modalità di interazione e la prima impressione saranno valutate tramite domande poste alla fine nel questionario.

4.2.3.3 Test sulle performance nella navigazione nell'ambiente

Alla sessione di training segue il test vero e proprio. Si tratta ovviamente di “primi usi” dell'interfaccia (*initial performance*) dato che non possiamo permetterci una valutazione delle prestazioni dell'utente dopo un costante uso dell'interfaccia, ma riteniamo che anche la sola misurazione delle prestazioni iniziali possano darci delle informazioni interessanti.

Un compito importante da testare, in quanto caratterizzante gli IL è quello della navigazione. Come è noto (Bowman et al. 2004), tale compito contiene dentro di sé due componenti: una parte fisica chiamata travel che include le singole azioni che l'utente compie per controllare la posizione e l'orientazione del punto di vista, una parte cognitiva chiamata wayfinding definita come il processo cognitivo del definire un percorso attraverso l'ambiente, usare ed acquisire conoscenza spaziale, aiutati da indizi sia di tipo naturale che artificiale.

In questa prima fase, si vuole testare l'efficacia in termini di usabilità delle tecniche di interazione utilizzate per la navigazione (travel) proponendo task che non includono una forte conoscenza spaziale o pianificazione, e cioè proponendo task di navigazione semplici verso oggetti la cui localizzazione è ovvia. Perché questo si verifichi il test viene svolto all'interno dello stesso ambiente semplice utilizzato per la sessione di training, un ambiente semplice e già conosciuto dall'utente. Dato che due sono le possibili modalità di navigazione verso un oggetto (navigazione libera o navigazione automatica su un oggetto), i task vengono fatti eseguire all'utente in entrambe le modalità.

Come misura per la valutazione si registra il tempo impiegato a svolgere l'operazione. Si prende inoltre nota di eventuali problemi, osservazioni e richieste di aiuto.

La lista dei task, e le metriche di valutazione sono riportate nelle seguenti tabelle, utilizzate dal valutatore per prendere nota dei risultati durante il test (Tabella 5, Tabella 6).

RISULTATI TEST TRAVEL (navigazione libera)		
Tasks	tempo	Note
Raggiungi l'immagine del crisantemo rosso.		
Segui il link che parte dal crisantemo rosso		
Raggiungi l'immagine del crisantemo blu		
Raggiungi la foto della margherita		
Raggiungi il testo intitolato "coltivazioni"		

Tabella 5 Tabella per la registrazione risultati test di travel in navigazione libera

RISULTATI TEST TRAVEL (navigazione tramite selezione)		
Tasks	tempo	Note
Raggiungi l'immagine del crisantemo rosso.		
Segui il link che parte dal crisantemo rosso		
Raggiungi l'immagine del crisantemo blu		
Raggiungi la foto della margherita		
Raggiungi il testo intitolato "coltivazioni"		

Tabella 6 Tabella per la registrazione risultati test di travel in navigazione per selezione

4.2.3.4 Testare la navigabilità negli IL: wayfinding

Il test sul wayfinding (orientamento) viene svolto in un ambiente più complesso rispetto a quello iniziale e cioè su un IL relativo alle arti grafiche lucchesi, progettato e realizzato anni fa presso il Laboratorio PERCRO (vedi capitolo 1, par 1.2.1).

Al fine di valutare il wayfinding si richiede all'utente di svolgere una libera esplorazione dell'ambiente per massimo 10 minuti. Durante questo tempo l'utente dovrà essere in grado di costruirsi una conoscenza spaziale (costruzione di una mappa interna dell'ambiente) che gli servirà successivamente, e di comprendere la metafora di organizzazione dell'informazione. Queste abilità saranno valutate alla fine dell'esplorazione.

Una prima modalità adottata per valutare il grado di conoscenza spaziale acquisita è quello di richiedere all'utente di esprimere con parole scritte o tramite disegni ciò che ha capito della strutturazione dell'ambiente appena visitato.

All'utente viene poi richiesto di compiere determinati task di navigazione verso determinati nodi target. Si è quindi cercato di individuare una lista di tasks di ricerca per il cui completamento (in modo efficiente) è necessario attingere alla propria conoscenza spaziale.

Come metriche di valutazione possiamo considerare il **tempo impiegato** all'esecuzione del task. Il valutatore potrà inoltre prendere nota delle eventuali **richieste di aiuto** da parte dell'utente e delle sue **reazioni**.

I task e le metriche sono riassunti nella tabella seguente (Tabella 7):

RISULTATI TEST WAYFINDING		
Tasks	tempo	Note
Trovare quali sono materiali e strumenti necessari a svolgere l'arte dell'acquerello		
Trovare informazioni su in che cosa consiste il procedimento litografico		
Trovare le opere litografiche presenti nel paesaggio		
Trovare informazioni su che cos'è la tecnica del disegno.		
Trova di nuovo in che cosa consiste il procedimento litografico (primed search)		

Tabella 7 Tabella per la registrazione risultati test di orientamento (wayfinding)

Ulteriori informazioni sul wayfinding saranno acquisite tramite domande poste all'interno del **questionario finale** volte a fornire informazioni più soggettive che possano servire a formulare ipotesi sui fattori determinanti i risultati ottenuti.

4.2.3.5 Questionario di valutazione

Come anticipato, alla fine del test sarà somministrato un questionario (appendice D) per valutare aspetti come il grado di accettazione generale dell'utente del sistema (user comfort e user satisfaction, cybersickness ecc), ma anche per approfondire alcuni punti (relativi ad esempio al travel e il wayfinding) che non potevano essere trattati durante la parte pratica del test. In appendice, è possibile prendere visione dei questionari somministrati (uno per la valutazione esperienza desktop e uno per quella immersiva).

4.2.3.6 Valutazione dei risultati

Alla fine del test avremo una serie di risultati: alcuni quantitativi e altri più qualitativi (feedback dell'utente). Nella prima tipologia di risultati rientrano:

- numero di tentativi effettuati nell'apprendimento dei singoli comandi di interazione nelle due
- tempi di apprendimento della tecnica di interazione nelle due modalità

- tempi nell'accesso alle informazioni in task di navigazione pura (quelli eseguiti nell'ambiente semplice) e in altri che implicano una maggiore capacità di orientamento (in ambiente più complesso), sempre per entrambe le modalità di fruizione

Nella seconda tipologia rientrano invece:

- feedback dell'utente attraverso il questionario che mirano soprattutto a fornire informazioni per la valutazione della soddisfazione
- feedback dell'utente durante il test (osservazioni, reazioni, risposte a eventuali richieste del valutatore).

In genere, per stabilire la bontà in termini di usabilità di un' applicazione a partire dai risultati ottenuti, vengono definiti dei requisiti minimi che l'applicazione deve avere affinché possa essere considerata usabile. È rispetto a tali requisiti minimi che i risultati dovrebbero essere confrontati. Tali requisiti vengono chiamati *specifiche di usabilità* e sono diversi a seconda dell'applicazione che stiamo valutando e del suo contesto di utilizzo.

Nel caso degli IL sicuramente alcuni aspetti importanti che un applicazione di questo tipo deve avere per poter essere considerata usabile sono:

- una buona velocità di apprendimento dell'interazione
- una buona velocità nel raggiungere le informazioni di cui abbiamo bisogno
- una buona soddisfazione nell'utente nell'utilizzo

Per quanto riguarda la velocità di apprendimento riteniamo che un tempo massimo di due minuti per l'apprendimento sia ragionevole. Riguardo invece alla velocità nel raggiungere le informazioni, è difficile quantificare numericamente quanto questa debba essere. Potremmo dire che negli ILs l'accesso all'informazione non dovrebbe essere superiore ai tempi di accesso necessari nei mezzi più tradizionali (libri, Internet ecc). Purtroppo però non siamo a conoscenza di studi in grado di fornirci queste misure di confronto. Inoltre forse il "giusto" tempo è quello che l'utente è disposto a considerare come tale; in altre parole, può darsi che l'utente di un IL sia disposto ad accettare maggiori tempi di accesso in cambio dei vantaggi che il nuovo mezzo fornisce. Per decidere quindi se i tempi di accesso alle informazioni rilevati sono buoni si terrà conto dell'opinione espressa dagli

utenti. Infine, in una scala da 1 a 5 (dove 1 indica scarso e 5 indica ottimo) riteniamo che perché l'applicazione possa essere considerata veramente soddisfacente, occorra ottenere almeno 4 come voto.

I risultati ottenuti per i due mezzi di fruizione saranno messi a confronto tra di loro, in modo da comprendere quale sia la modalità più appropriata per l'utilizzo dei Paesaggi di Informazione.

4.2.4 Scelta degli utenti e loro distribuzione nell'esperimento

Poiché l'esperimento deve fornire una valutazione dell'usabilità degli Information Landscapes sia in ambiente desktop che immersivo, si è posto il problema (non semplice) di decidere come distribuire gli utenti tra i due esperimenti.

Per la prima parte dell'esperimento - quella relativa al training e all'uso del dispositivo di interazione - uno stesso utente può svolgere entrambe le esperienze senza problemi dato che si può assumere che non vi sia una grande influenza dalla prima alla seconda esperienza⁴⁰. Per la parte relativa al test di orientamento, non è possibile utilizzare uno stesso utente per entrambe le modalità, perché la conoscenza acquisita durante la prima esperienza - l'aver cioè già esplorato l'ambiente e aver svolto in esso i tasks - influenzerebbe la successiva esperienza.

Quindi, si è reso necessario l'istituzione di due gruppi di utenti distinti. Una possibilità poteva essere quella di far svolgere al primo gruppo il test solo su PC e al secondo gruppo il test solo nel cave. Si è invece deciso di far svolgere entrambe le esperienze ai due gruppi ma in ordine invertito (Tabella 8).

	I gruppo	II gruppo
Esperienza 1	Computer	Cave
Esperienza 2	Cave	computer

Tabella 8 Modalità di assegnazione degli utenti nel test

⁴⁰ L'ambiente è infatti molto semplice e conosciuto bene dall'utente sia nella prima che nella seconda esperienza dato che è in questo ambiente che si svolge il training iniziale. In questa parte del test, ci concentriamo infatti sulle performance nell'utilizzo del dispositivo di interazione per compiti di navigazione, e non sulla conoscenza dell'ambiente.

In questo modo, per la prima parte dell'esperimento possiamo ottenere una maggiore quantità di risultati dato che ogni utente svolge il test in entrambe le modalità, mentre per la parte dell'esperimento relativa al test di orientamento confrontiamo solamente i dati relativi alla prima esperienza.

Per il test di orientamento si è deciso comunque di sfruttare anche i dati relativi alla seconda esperienza per un confronto incrociato permettendoci di valutare la trasferibilità della conoscenza da un contesto di esecuzione all'altro (Tabella 9).

	I gruppo	II gruppo	
Esp.1	Computer	Cave	← Direttamente confrontabili
Esp. 2	Cave	computer	

Tabella 9 Modo in cui saranno considerati i dati in fase di analisi

Inoltre, si è stabilito che ogni gruppo sia costituito da 6 persone. Come specificato in Hix et al. i partecipanti dovrebbero rappresentare i tipi di categorie di utenti dell'interfaccia che andiamo valutando, dovrebbero cioè accordarsi a specifici attributi come background generale, grado di abilità, conoscenza del computer, conoscenza del dominio dell'applicazione e così via. Nel nostro caso, per ogni gruppo sono stati considerati due sottogruppi di utenti: un sottogruppo di 3 utenti con esperienza di uso di applicazioni tridimensionali interattive e un sottogruppo di 3 utenti con una buona alfabetizzazione informatica di base ma senza esperienza con applicazioni 3D.

Riassumendo, abbiamo due gruppi di 6 utenti (di cui 3 esperti e 3 inesperti) per un totale di 12 utenti. Il primo gruppo svolge l'esperienza descritta nel paragrafo 4.2.3 prima su PC e poi nel CAVE, il secondo gruppo svolge l'esperienza prima su CAVE e poi su PC.

4.2.5 Descrizione delle due modalità di fruizione

Le modalità di fruizione sono quindi due: quella desktop e quella immersiva. Nella modalità desktop si utilizza un PC portatile ASUS N53S avente le seguenti caratteristiche tecniche:

- CPU Intel Core I7 - 2670M, 2.2GHz
- RAM: 8 GB
- Scheda grafica: NVIDIA GEFORCE GT 630M (2GB)
- sistema operativo: Windows 7 Premium

L'interazione è avvenuta tramite un mouse secondo le seguenti modalità:

- pressione del bottone sinistro e movimento verticale per ottenere lo spostamento in avanti e indietro della camera nell'ambiente virtuale.
- pressione del bottone sinistro e movimento in orizzontale per la rotazione della vista a destra e sinistra
- pressione del bottone destro del mouse e movimento in orizzontale per traslare la posizione della camera verso destra o verso sinistra
- pressione del bottone destro del mouse e movimento in verticale per traslare la posizione della camera verso l'alto e verso il basso
- click con il tasto destro su un oggetto per la navigazione automatica verso di esso
- click con il tasto sinistro su un oggetto con link per attivare il collegamento
- per tornare automaticamente alla posizione iniziale si utilizza un bottone esterno.

Nella modalità immersiva la fruizione è avvenuta all'interno del X-CAVE presente presso il laboratorio PERCRO della Scuola Superiore S.Anna. Si tratta di una stanza di 16 metri quadrati in cui pareti e pavimento (ad eccezione del soffitto e di una parete) consistono in superfici proiettate, permettendo a chi vi staziona dentro di essere avvolto dalla grafica 3D e di interagire con ambienti simulati al computer. La sua architettura è particolarmente complessa: le immagini 3D sono generate da 18 proiettori hd-ready, controllati a loro volta da 6 workstation grafiche ad alte prestazioni per una risoluzione

totale di oltre 16 milioni di pixels, una luminosità complessiva di 54000 Ansi Lumen e un volume di visualizzazione di oltre 40 metri cubi.

Per l'interazione si è utilizzato un joystick wireless Thrustmaster (Figura 44) che si è programmato nel seguente modo:

- per andare in avanti e indietro si sposta la levetta di sinistra in alto e in basso
- per ruotare la camera a sinistra o a destra si sposta la levetta di destra a sinistra e a destra
- per andare in alto o in basso si tiene premuto contemporaneamente uno dei tasti frontali del joystick, e si sposta la levetta di sinistra verso l'alto o verso il basso.
- Per selezionare un oggetto, esso va portato al centro dello schermo. Per facilitare il compito è stato inserito un mirino: l'oggetto che si intende selezionare deve trovarsi sotto il mirino.
- Per attivare la navigazione automatica verso l'oggetto selezionato si utilizza il bottone 1 del joystick.
- Per seguire un link presente su un oggetto selezionato si utilizza il bottone 2.
- per tornare automaticamente alla posizione iniziale, si utilizza il bottone con il simbolo della casa posto centralmente nel joystick.



Figura 44 JOYSTICK WIRELESS THRUSTMASTER

4.3 Risultati dell'esperimento

4.3.1 Apprendimento della tecnica di interazione con il mouse

La tabella seguente (Tabella 10) riporta il numero medio di tentativi effettuati dagli utenti prima di essere in grado di effettuare ogni comando utilizzando il mouse su PC. Il dato è riportato per gli utenti esperti, per quelli inesperti, e per tutti gli utenti considerati complessivamente. Si riporta inoltre una rappresentazione grafica degli stessi dati (Figura 50).

	Media del numero di tentativi su PC		
	<i>Utenti esperti</i>	<i>Utenti inesperti</i>	<i>Esperti + inesperti</i>
Avanti	1,67	2,50	2,08
Indietro	1,00	1,67	1,33
Ruota sx	1,00	1,83	1,42
Ruota dx	1,33	1,67	1,50
destra	1,50	1,33	1,42
Sinistra	1,00	1,50	1,25
Sopra	1,00	1,17	1,08
Sotto	1,00	1,17	1,08
Seguire link	1,17	1,00	1,08
Navigazione automatica	1,33	1,50	1,42

Tabella 10 Numero di tentativi effettuati per l'apprendimento di ogni azione con mouse

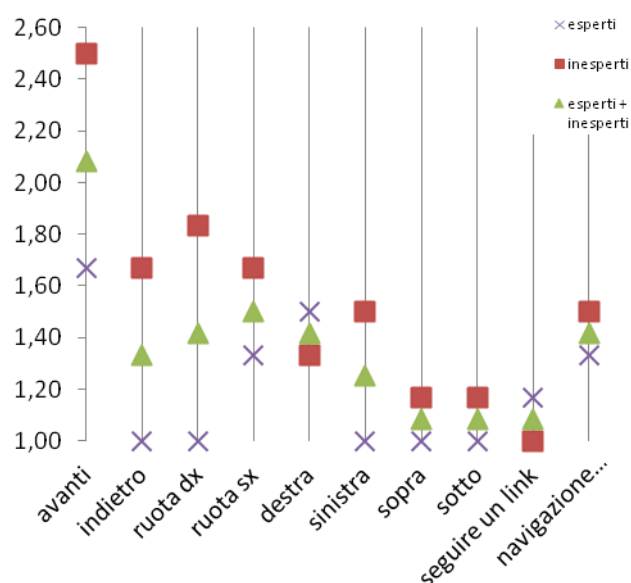


Figura 50 Rappresentazione grafica del numero di tentativi per l'apprendimento di ogni azione (mouse) da parte di ogni categoria di utente

In generale il numero medio di tentativi nell'apprendimento delle singole azioni è piuttosto basso, sempre al di sotto di 2,5 tentativi. Come era prevedibile, nella maggior parte dei casi gli utenti inesperti impiegano mediamente più tentativi rispetto a quelli esperti nell'apprendimento delle singole azioni. Secondo questi dati le azioni più difficili da imparare sono quella relativa ad andare avanti, indietro e la rotazione della vista.

Il tempo complessivo medio dedicato all'apprendimento della tecnica di interazione è intorno ai 53 secondi per l'utente esperto, circa il doppio per l'utente inesperto, e globalmente di circa 78 secondi, come riportato in tabella (Tabella 11).

	Tempo medio di apprendimento su PC (in sec)
Utenti esperti	53,67
Utenti inesperti	103,83
Utenti esperti + inesperti	78,75

Tabella 11 Tempi medi di apprendimento della tecnica di interazione

Possiamo affermare che il tempo di apprendimento è oggettivamente accettabile in riferimento al requisito di immediatezza che un IL deve avere, ma è importante anche capire il pensiero degli utenti relativamente alla difficoltà e tempi di apprendimento.

Per questo motivo, oltre ai dati oggettivi appena riportati, dal questionario siamo in grado di rilevare la percezione soggettiva degli utenti relativamente alla difficoltà nell'apprendimento della tecnica di interazione (Tabella 12)

	Utenti esperti	Utenti inesperti	Tutti gli utenti
No, nessuna particolare difficoltà	17%	33%	25%
Si, alcune difficoltà	83%	50%	67%
Si, molte difficoltà	0%	17%	8%

Tabella 12 Risultati domanda: "Hai incontrato difficoltà nell'apprendimento della tecnica di utilizzo del mouse per interagire con l'ambiente?"

Sia nel caso degli utenti esperti che in quello degli inesperti, la maggior parte ha risposto di aver incontrato qualche difficoltà. Inaspettatamente però si riscontra un numero maggiore di utenti esperti. Gli utenti esperti motivano la loro difficoltà nell'abitudine ad usare altre combinazioni di tasti, l'eccessiva sensibilità del mouse specialmente nell'effettuare le rotazioni, nel fatto che le due possibili tecniche di navigazione nell'ambiente - quella libera, e quella automatica - non siano integrate correttamente, nella difficoltà a memorizzare le funzioni dei tasti. Anche gli utenti inesperti notano una eccessiva sensibilità nel mouse (spostamenti troppo veloci rispetto ai movimenti della mano), il fatto che i vari comandi possano essere confusi facilmente a causa della presenza di pochi tasti e di numerose combinazioni, il fatto di non aver mai usato il mouse per interagire con un ambiente 3D. È probabile che gli utenti esperti siano stati più duri nel giudizio per il fatto che sono abituati ad usare il mouse in applicazioni 3D ma magari secondo combinazioni differenti, per il fatto di riuscire a notare più problemi, o perché erano consapevoli che fosse possibile gestire meglio l'interazione con il mouse. Gli utenti inesperti in genere sono più portati a ritenersi responsabili o colpevolizzarsi delle difficoltà incontrate.

La maggior parte degli utenti esperti, inoltre ritiene abbastanza accettabile il tempo necessario all'apprendimento della tecnica di interazione, a differenza degli utenti inesperti il cui giudizio è meno positivo. Considerando globalmente tutti gli utenti, il voto è di 3,2 su 5 (Tabella 13)

	media voti
esperti	3,5
inesperti	2,8
tutti	3,2

Tabella 13 Risultati domanda: "Quanto ritieni accettabile il tempo necessario all'apprendimento della tecnica di interazione con il mouse, in una scala da 1 a 5?"

4.3.2 Apprendimento della tecnica di interazione con il joystick

La tabella 14 riporta il numero medio di tentativi effettuati dagli utenti prima di essere in grado di effettuare ogni comando utilizzando il joystick all'interno del cave. Il dato è riportato per gli utenti esperti, per quelli inesperti, e per tutti gli utenti considerati complessivamente. Si riporta inoltre una rappresentazione grafica degli stessi dati (Figura).

	Media del numero di tentativi nel CAVE		
	Utenti esperti	Utenti inesperti	Esperti + inesperti
Avanti	1,17	1,17	1,17
Indietro	1,17	1,17	1,17
Ruota sx	1,00	1,17	1,08
Ruota dx	1,00	1,17	1,08
destra	1,00	1,00	1,00
Sinistra	1,00	1,00	1,00
Alto	1,17	2,00	1,58
Basso	1,17	2,00	1,58
Seguire link	1,33	1,00	1,17
Navigazione automatica	1,00	1,00	1,00

Tabella 14 Numero di tentativi effettuati per l'apprendimento di ogni azione con il joystick

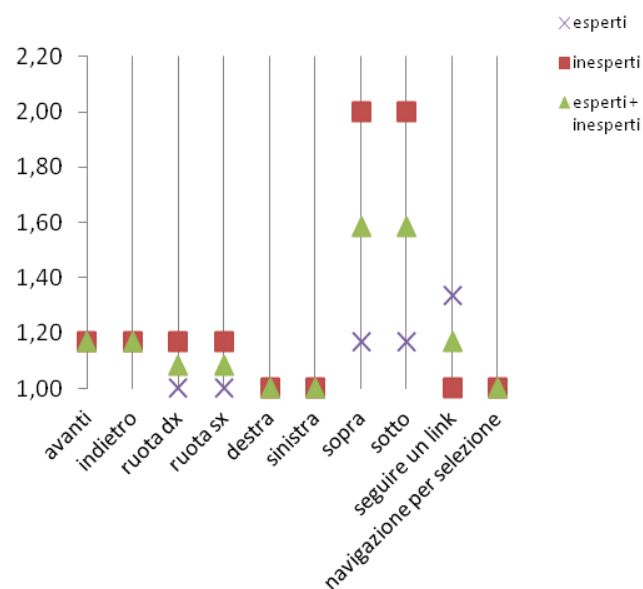


FIGURA 51 Rappresentazione grafica del numero di tentativi per l'apprendimento di ogni azione (joystick) da parte di ogni categoria di utente

Come mostrato dal grafico, non si riscontra un grande differenza tra i risultati degli utenti esperti e quelli inesperti tranne che nell'azione di andare in alto e in basso. Tale azione è

effettivamente più complessa da svolgere rispetto alle altre dato che richiede contemporaneamente la pressione di un tasto frontale del joystick e l'utilizzo della levetta di sinistra.

Il tempo complessivo medio dedicato all'apprendimento della tecnica di interazione è intorno ai 70 secondi per l'utente esperto, intorno ai 107 secondi per l'utente inesperto, e globalmente di circa 88 secondi, come riportato in tabella 15.

	Tempo medio di apprendimento su PC (in sec)
Utenti esperti	69,67
Utenti inesperti	106,67
Utenti esperti + inesperti	88,16

Tabella 15 tempo medio di apprendimento della tecnica di interazione con joystick nel cave

Anche in questo caso, i tempi sono al di sotto di quanto stabilito dalla specifica di usabilità, per cui sono accettabili.

Da quanto emerge dal questionario, la maggior parte degli utenti sia esperti che inesperti non ha percepito una particolare difficoltà nell'apprendimento della tecnica di interazione con il joystick (Tabella 16).

	Utenti esperti	Utenti inesperti	Tutti gli utenti
No, nessuna particolare difficoltà	83%	67%	75%
Si, alcune difficoltà	17%	33%	25%
Si, molte difficoltà	0%	0%	0%

Tabella 16 risultati domanda: "Hai incontrato difficoltà nell'apprendimento della tecnica di utilizzo del joystick per interagire con l'ambiente?"

Le poche persone che hanno trovato qualche difficoltà le attribuiscono alla fatica a ricordarsi tutte le funzionalità del joystick, in particolare la combinazione di tasti per andare in alto e in basso, e alla poca dimestichezza nel maneggiare il joystick.

Anche per quanto riguarda il tempo necessario all'apprendimento, la maggior parte degli utenti ha gradito molto la tecnica di interazione con il joystick (Tabella 17).

	media voti
esperti	4,8
inesperti	4,0
tutti	4,4

Tabella 17 Risultati domanda: "Quanto ritieni accettabile il tempo necessario all'apprendimento della tecnica di interazione con il mouse, in una scala da 1 a 5?"

4.3.3 Apprendimento della tecnica di interazione: confronto tra mouse e joystick

Se mettiamo a confronto i risultati relativi al mouse con quelli relativi al joystick ci rendiamo conto come, generalmente, il numero medio di tentativi necessari per apprendere le singole azioni sia maggiore nel caso del mouse, fatta eccezione per l'apprendimento della combinazione per andare in alto e in basso e per seguire un link (Figura).

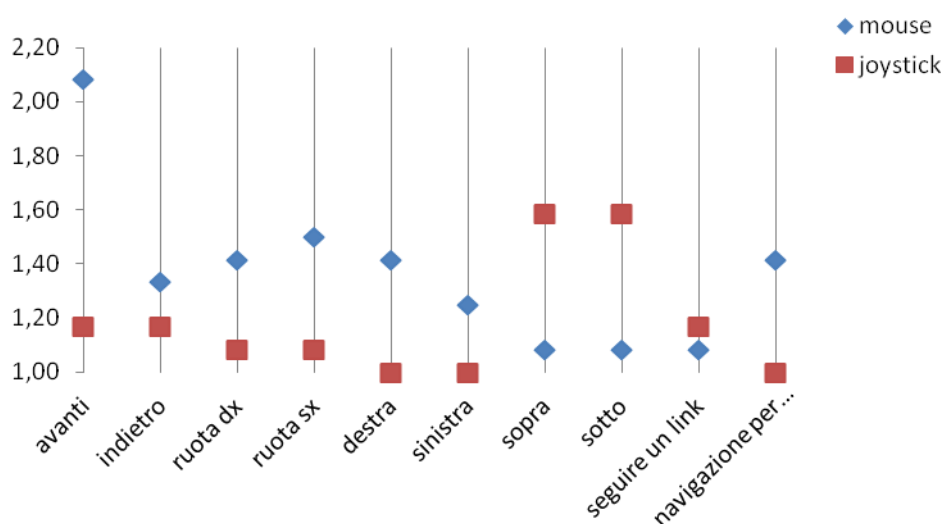


Figura 52 Confronto tra numero di tentativi effettuati per imparare le singole azioni con mouse e con joystick

I tempi dedicati all'apprendimento in media sono più o meno gli stessi, se pur lievemente superiori per il joystick (Figura).

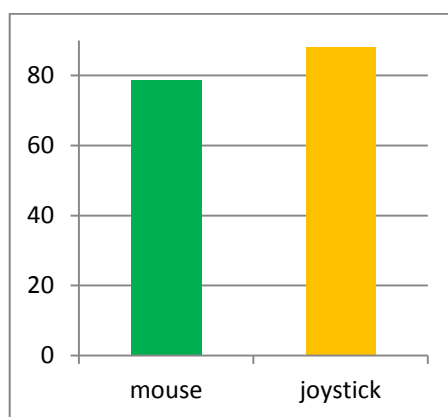


Figura 53 Confronto tra tempi medi di apprendimento della tecnica di interazione con MOUSE E CON JOYSTICK

Gli utenti in generale riferiscono di aver trovato difficoltà nell'apprendimento della tecnica di interazione con il mouse, mentre se la sono cavata bene nell'apprendimento della tecnica con il joystick. La percentuale di partecipanti che hanno espresso un giudizio

positivo sul tempo necessario all'apprendimento è superiore per il joystick rispetto al mouse. I giudizi sono maggiormente negativi nel caso degli utenti inesperti, che per forza di cose hanno incontrato più difficoltà nell'apprendimento.

4.3.4 Test di navigazione (travel) su PC con mouse

Le tabelle seguenti (Tabella 18,19,20) riassumono i tempi medi impiegati dagli utenti nell'esecuzione di ciascun task di navigazione su PC. Il risultato è in linea con quanto ci si aspettava. La navigazione automatica permette di raggiungere più velocemente gli elementi. Quando detto vale anche analizzando individualmente i tempi degli utenti esperti e di quelli inesperti, anche se questi ultimi hanno in genere tempi maggiori rispetto agli utenti inesperti.

	Media dei tempi (in sec) Tutti gli utenti	
	<i>Navigazione tradizionale</i>	<i>Navigazione automatica</i>
Task1	14,33	12,75
Task2	7,92	5,83
Task3	45,67	13,92
Task4	66,08	36,33
Task5	28,50	13,00

Tabella 18 tempi medi nei task di navigazione su pc (tutti gli utenti)

	Media dei tempi (in sec) degli utenti esperti	
	<i>Navigazione tradizionale</i>	<i>Navigazione automatica</i>
Task1	12,50	10,67
Task2	8,33	5,17
Task3	19,17	10,67
Task4	43,33	16,17
Task5	20,50	8,50

Tabella 19 tempi medi nei task di navigazione su pc(utenti esperti)

	Media dei tempi (in sec) degli utenti inesperti	
	<i>Navigazione tradizionale</i>	<i>Navigazione automatica</i>
Task1	16,17	14,83
Task2	7,50	6,50
Task3	72,17	17,17
Task4	88,83	56,50
Task5	36,50	17,50

Tabella 20 tempi medi nei task di navigazione su pc (utenti inesperti)

In generale i tempi sono abbastanza bassi. Il task in cui gli utenti esperti hanno durato molta più fatica rispetto a quelli esperti è il posizionamento dietro ad un oggetto.

Gli utenti esperti in genere non si sono lamentati del tempo che hanno impiegato nello svolgimento dei task, anche se ritenevano che eventuali miglioramenti della tecnica di interazione avrebbero potuto portare a migliori tempi; alcuni utenti inesperti hanno invece avuto l'impressione di metterci troppo e hanno attribuito questo fatto alla loro inesperienza.

4.3.5 Test di navigazione (travel) nel CAVE con joystick

Anche nell'interazione con il joystick, la navigazione automatica ha determinato nella maggior parte dei casi dei tempi inferiori rispetto alla navigazione libera (Tabella 21) Fa eccezione il task 1, che richiedeva dalla posizione iniziale di raggiungere l'immagine di un elemento che si trovava dietro un altro elemento. In tal caso l'utente, chiamato ad utilizzare la navigazione automatica, ha dovuto mettersi in posizione favorevole alla selezione dell'oggetto prima di poter attivare il comando di navigazione automatica. Questi risultati sono veri anche considerando distintamente il gruppo di utenti esperti e inesperti (Tabella 22, Tabella 23), anche se questi ultimi hanno avuto dei tempi maggiori.

	Media dei tempi (in sec)	
	<i>Navigazione tradizionale</i>	<i>Navigazione automatica</i>
Task1	6,25	13,00
Task2	9,00	5,00
Task3	19,25	13,33
Task4	23,58	12,83
Task5	20,33	9,17

Tabella 21 tempi medi nei task di navigazione nel cave (tutti gli utenti)

	Media dei tempi (in sec) degli utenti esperti	
	<i>Navigazione tradizionale</i>	<i>Navigazione automatica</i>
Task1	7	13,00
Task2	6,83	5,00
Task3	15,50	13,33
Task5	13,00	12,83
Task6	12,17	9,17

Tabella 22 tempi medi nei task di navigazione nel cave (utenti esperti)

	Media dei tempi (in sec) (inesperti)	
	<i>Navigazione tradizionale</i>	<i>Navigazione automatica</i>
Task1	5,50	18,83
Task2	11,17	5,33
Task3	23,00	19,33
Task4	34,17	19,50
Task5	28,50	21,67

Tabella 23 tempi medi nei task di navigazione nel cave (utenti inesperti)

Nessun utente ha lamentato di aver impiegato troppo tempo nel compiere i task di navigazione.

4.3.6 Test di navigazione: confronto tra mouse e joystick

Allo scopo di facilitare il confronto riportiamo un'unica tabella i dati già esaminati separatamente sopra (Tabella 24):

	Media dei tempi (in sec)			
	<i>Navigazione tradizionale (mouse)</i>	<i>Navigazione tradizionale (joystick)</i>	<i>Navigazione automatica (mouse)</i>	<i>Navigazione automatica (joystick)</i>
Task1	14,33	6,25	12,75	13,00
Task2	7,92	9,00	5,83	5,00
Task3	45,67	19,25	13,92	13,33
Task4	66,08	23,58	36,33	12,83
Task5	28,50	20,33	13,00	9,17

Tabella 24 confronto tra i tempi impiegati nei task di navigazione su pc e nel cave (tutti gli utenti) per le due modalità di navigazione

Per quanto riguarda la navigazione tradizionale, l'impiego del joystick si è rivelato notevolmente più veloce in 4 task su 5. Per quanto riguarda la navigazione automatica, l'impiego del joystick si è rivelato lievemente più veloce rispetto al mouse in 4 task su 5. Quest'ultimo dato un po' stupisce: ci saremmo aspettati il contrario dato che con il mouse la selezione di un oggetto è più immediata dovendo solo portare il puntatore del mouse su di esso, mentre nel caso del joystick occorre portare l'oggetto al centro dello schermo per la selezione.

4.3.7 Test di orientamento

Come detto sopra, abbiamo due gruppi di utenti, il primo che ha effettuato prima l'esperienza desktop e poi quella nel cave, e il secondo che ha svolto prima l'esperienza nel cave e poi nel pc. Agli utenti è stato chiesto di navigare l'ambiente *CubiSpace* per massimo 10 minuti, durante i quali hanno dovuto costruirsi una mappa spaziale dell'ambiente da descrivere successivamente su un foglio a parole e/o disegni. Successivamente è stato chiesto loro di svolgere alcuni task di orientamento.

Prima esperienza (desktop e cave)

Esaminiamo adesso i risultati relativi alla prima esperienza desktop e immersiva.

Gli utenti desktop hanno ottenuto una discreta comprensione dell'organizzazione delle informazioni nell'ambiente. Quasi tutti hanno capito la presenza di 4 aree tematiche relative ad una determinata tecnica artistica, hanno compreso che queste sono rappresentate da cubi collocati nello spazio e che ogni cubo contiene su ogni faccia tipologie diverse di contenuti relative alla tecnica in oggetto. Inoltre hanno notato la presenza al centro di una specie di menu/legenda che rimanda tramite link ai vari cubi. Di seguito si riportano alcune descrizioni fornite dagli utenti:

"Ho quattro argomenti disposti su 4 cubi ai quali posso arrivare grazie a 4 link iniziali. In ogni cubo trovo link o immagini"

"4 laboratori rappresentati attraverso dei cubi. 4 facce dei cubi indicano: opere, materiali, storia, ecc. Disposti nello spazio con al centro una legenda. I 4 laboratori sono linkati da un sommario che li riferisce tutti".

Tuttavia, si noti come queste descrizioni non facciano menzione della distribuzione spaziale dei cubi nello spazio. Soltanto una persona è stata in grado di produrre un disegno della disposizione dei cubi nello spazio. Gli altri hanno riferito di non ricordare la disposizione spaziale dei cubi.

Molti degli utenti che invece hanno fatto la loro prima esperienza nel cave oltre ad essere in grado di descrivere la logica organizzativa delle informazioni, sono stati in grado anche di riferire la posizione spaziale dei cubi. Alcuni utenti ricordavano addirittura la tematica associata a ciascun cubo (Figura 54).

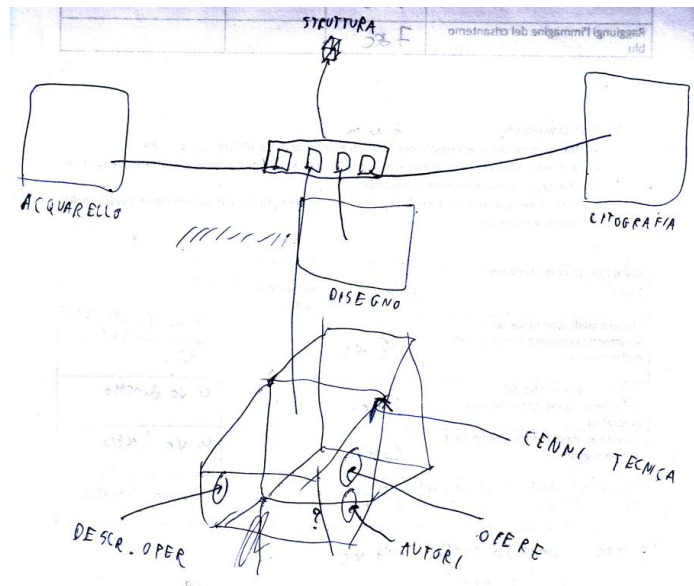


Figura 54 Schizzo realizzato da un utente dopo la navigazione di "Cubispace" nel cave

La causa potrebbe essere il maggior campo di vista (FOV) che il cave fornisce, mentre lo schermo limita il campo di vista ad una visione frontale e limitata, rimandando all'utente il compito di ricostruire nella sua mente una visione complessiva del paesaggio durante la navigazione. Alcuni utenti segnalano inoltre come i link abbiano maggior senso nel cave che su schermo e aiutino a comprendere la struttura spaziale delle informazioni: il FOV limitato su schermo impedisce di vedere la destinazione del link, quando questa si trova lontana dalla sorgente.

Agli utenti è stato inoltre richiesto di svolgere determinati compiti di ricerca nell'ambiente allo scopo di valutare l'orientamento spaziale. I task sono gli stessi sia per il gruppo desktop che per il gruppo cave, in modo da poter effettuare un confronto tra i risultati (Tabella 25).

	Task1	Task2	Task3	Task4	Task5
Medie dei tempi (sec) modalità desktop	25,33	88,17	35,50	86,00	62,50
Medie dei tempi (sec) modalità immersiva	27,17	41,00	39,17	53,50	27,33

Tabella 25 Confronto tra i tempi impiegati in modalità desktop e in modalità immersiva nei task di orientamento (prima esperienza)

In 3/5 tasks la combinazione cave+joystick si è rivelata notevolmente più veloce di quella desktop+mouse, in 2 task le prestazioni con i due dispositivi si sono più o meno eguagliate. Per motivi di spazio non abbiamo riportato i dati relativi ad ogni categoria di utente separatamente. Si fa presente che comunque, anche se in genere gli utenti esperti sono più veloci rispetto agli utenti inesperti nel risolvere task di orientamento, questo non è sempre vero. Questo sembra suggerire il fatto che i task di orientamento non sono tanto legati al livello di expertise nell'uso del dispositivo ma ad una predisposizione individuale.

Tuttavia, i risultati ottenuti non consentono di affermare con certezza che l'orientamento sia favorito in ambiente immersivo rispetto ad ambiente desktop, dato che come abbiamo visto sopra, l'interazione con il joystick si è dimostrata nettamente più veloce di quella col mouse nella maggior parte dei tasks. Per stabilire se veramente la differenza dei tempi è dovuta al fattore orientamento forse sarebbe opportuno eseguire il test sia su computer che su cave utilizzando un dispositivo di interazione comune, per esempio il joystick⁴¹. Per il momento l'unica cosa che possiamo affermare è che anche in task che implicano una necessità di sapersi orientare l'utente ha impiegato in molti casi tempi inferiori con il joystick nel cave rispetto al mouse su schermo.

Focalizzando l'attenzione sulle metodologie di esplorazione utilizzate dagli utenti per il raggiungimento degli obiettivi, si nota inoltre come pressoché tutti gli utenti desktop hanno utilizzato il sistema di navigazione con i link, mentre alcuni utenti della modalità immersiva raggiungevano direttamente la locazione desiderata senza bisogno di utilizzare i link che il paesaggio metteva a disposizione.

⁴¹ Per questo motivo si è deciso successivamente di far eseguire l'esperimento di orientamento ad un gruppo di 6 utenti (3 esperti, 3 inesperti) su PC con il joystick. Mettendo a confronto i risultati di questi utenti con quelli ottenuti dai 6 utenti che avevano utilizzato il joystick nel cave, è stato provato che nella maggior parte dei task (4/5) gli utenti desktop hanno impiegato molto più tempo.

	<i>Task1</i>	<i>Task2</i>	<i>Task3</i>	<i>Task4</i>	<i>Task5</i>
Tempi (sec) modalità desktop	53,67	53,83	20,33	73,00	45,83
Tempi (sec) modalità immersiva	27,17	41,00	39,17	53,50	27,33

In questo modo abbiamo effettivamente provato che l'orientamento è favorito dalla modalità immersiva rispetto a quella desktop.

Riportiamo il feedback ricevuto sul questionario relativamente all'orientamento nell'IL Cubispace. Nella modalità desktop una buona percentuale ritiene di aver avuto molte difficoltà nell'esplorazione dell'ambiente, mentre nella modalità immersiva sono segnalate solo alcune difficoltà anche se diffuse tra gli tutti gli utenti (Tabella 26). Un dato che stupisce è il fatto che nella modalità desktop vi sono utenti che segnalano di non aver avuto alcuna difficoltà mentre nella modalità immersiva tutti ritengono di aver incontrato qualche difficoltà.

Modalità desktop

	<i>tutti utenti</i>	<i>esperti</i>	<i>inesperti</i>
qualche difficoltà	17%	33%	0%
nessuna rilevante difficoltà	50%	67%	33%
molte difficoltà	33%	0%	67%

Modalità immersiva

	<i>tutti utenti</i>	<i>esperti</i>	<i>inesperti</i>
qualche difficoltà	100%	100%	100%
nessuna rilevante difficoltà	0%	0%	0%
molte difficoltà	0%	0%	0%

Tabella 26 Risultati domanda : " Ritieni di aver avuto difficoltà durante l'esplorazione dell'ambiente?"

Le maggiori difficoltà riportate dagli utenti per entrambe le modalità sono riportate nel grafico (Figura 55). In entrambe la modalità, esse riguardano l'uso del dispositivo per la navigazione e la mancanza di esperienza. Due dati che però stupiscono sono: 1) il fatto che vi siano molti utenti nella modalità immersiva che avvertono la necessità di facilitazioni per l'orientamento nello spazio e non ve ne siano nella modalità desktop; 2) il fatto che siano di più nella modalità immersiva gli utenti che ritengono che le informazioni non siano organizzate in modo chiaro nell'ambiente.

È probabile che la ragione di queste stranezze risieda nel fatto che il numero di partecipanti all'esperimento è relativamente basso.

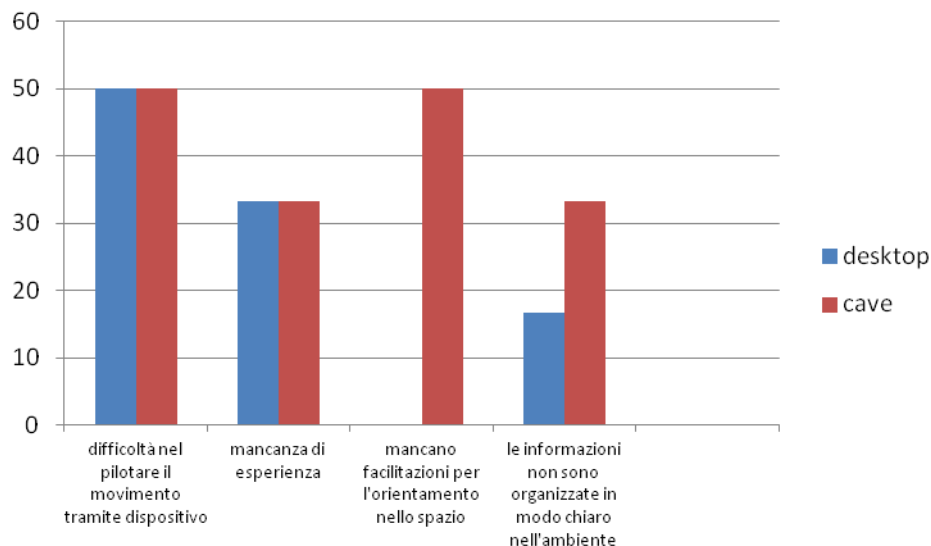


Figura 55 Difficoltà incontrate nell'esplorazione di cubispace (confronto tra modalità desktop e immersiva)

Nella modalità desktop una buona percentuale di utenti segnala di essersi perso per molte volte, mentre nella modalità immersiva chi si è perso lo ha fatto solo qualche volta (Tabella 27).

Modalità desktop

	<i>tutti utenti</i>	<i>esperti</i>	<i>inesperti</i>
qualche volta	33%	33%	33%
molte volte	33%	0%	67%
mai	33%	67%	0%

Modalità immersiva

	<i>tutti utenti</i>	<i>esperti</i>	<i>inesperti</i>
qualche volta	67%	67%	67%
molte volte	0%	0%	0%
mai	33%	33%	33%

Tabella 27 Risultati domanda: "Quante volte ti è capitato di sentirti perso?"

Sia nella modalità desktop che immersiva la maggioranza degli utenti ritiene che la presenza di una mappa spaziale avrebbe potuto facilitare loro l'orientamento anche se la richiesta è stata maggiore da parte degli utenti nella modalità desktop (Tabella 28).

Modalità desktop

	<i>tutti utenti</i>	<i>esperti</i>	<i>inesperti</i>
sicuramente	83%	100%	67%
forse	17%	0%	33%
non credo	0%	0%	0%

Modalità immersiva

	<i>tutti utenti</i>	<i>esperti</i>	<i>inesperti</i>
sicuramente	50%	67%	33%
forse	33%	33%	33%
non credo	17%	0%	33%

Tabella 28 Risultati domanda: "Ritieni che la presenza di una mappa spaziale avrebbe potuto facilitarti l'orientamento?"

Tutti gli utenti desktop ritengono di aver compreso soltanto in parte la struttura delle informazioni all'interno dell'ambiente, mentre una buona porzione di utenti nella modalità immersiva ritengono di aver raggiunto una comprensione totale (Tabella 29).

Modalità desktop

	<i>tutti utenti</i>	<i>esperti</i>	<i>inesperti</i>
per niente	0%	0%	0%
in parte	100%	100%	100%
totalmente	0%	0%	0%

Modalità immersiva

	<i>tutti utenti</i>	<i>esperti</i>	<i>inesperti</i>
per niente	0%	0%	0%
in parte	33%	33%	33%
totalmente	67%	67%	67%

Tabella 29 Risultati domanda: "Ritieni di aver compreso la struttura spaziale delle informazioni?"

Al fine di conoscere il background delle persone che hanno partecipato all'esperimento è stato chiesto il livello di conoscenza della tematica trattata nell'IL e cioè l'arte. La maggior parte degli utenti desktop (83%) che ha partecipato all'esperimento ha segnalato di non saper niente sull'argomento mentre nella modalità immersiva molti più utenti sapevano qualcosa sull'argomento (Tabella 30).

Modalità desktop

	<i>tutti utenti</i>	<i>esperti</i>	<i>inesperti</i>
non so niente sull'argomento	83%	67%	100%
ne so qualcosa	17%	33%	0%
sono un esperto di questo settore	0%	0%	0%

Modalità immersiva

	<i>tutti utenti</i>	<i>esperti</i>	<i>inesperti</i>
non so niente sull'argomento	33%	33%	33%
ne so qualcosa	67%	67%	67%
sono un esperto di questo settore	0%	0%	0%

Tabella 30 Risultati domanda: "Eri già a conoscenza da altre fonti della tematica affrontata nell'IL?"

Seconda esperienza

Come specificato sopra, agli utenti è stato richiesto di svolgere di nuovo i task nell'altra modalità allo scopo di effettuare un confronto incrociato.

Confrontando i tempi impiegati dagli utenti che hanno fruito della modalità cave come prima esperienza, con quelli ottenuti dagli utenti che hanno effettuato l'esperienza cave dopo quella desktop, i tempi sono più o meno gli stessi fatta eccezione per il task3 (Tabella 31).

	<i>Task1</i>	<i>Task2</i>	<i>Task3</i>	<i>Task4</i>	<i>Task5</i>
Medie dei tempi modalità immersiva (PRIMA ESPERIENZA)	76,50	41,00	39,17	53,50	27,33
Medie dei tempi modalità immersiva (SECONDA ESPERIENZA)	76,00	48,00	18,17	51,50	27,50

Tabella 31 Confronto tra i tempi impiegati dagli utenti che hanno fruito cubispace nella modalità cave come prima esperienza, con quelli ottenuti dagli utenti che hanno effettuato l'esperienza cave dopo quella desktop

Invece, i tempi degli utenti che hanno sperimentato l'IL sul computer dopo aver fatto esperienza nel cave, hanno riportato in diversi casi tempi notevolmente inferiori rispetto a chi ha usato il computer alla prima esperienza (Tabella 32).

	<i>Task1</i>	<i>Task2</i>	<i>Task3</i>	<i>Task4</i>	<i>Task5</i>
Medie dei tempi (sec) modalità desktop (PRIMA ESPERIENZA)	25,33	88,17	35,50	86,00	62,50
Medie dei tempi (sec) modalità desktop (SECONDA ESPERIENZA)	24,67	46,17	36,00	61,17	36,67

Tabella 32 Confronto tra i tempi impiegati dagli utenti che hanno fruito cubispace nella modalità DESKTOP come prima esperienza, con quelli ottenuti dagli utenti che hanno effettuato l'esperienza DESKTOP dopo quella IMMERSIVA

Questo è in linea con quanto ipotizzato. Sembra che la modalità immersiva favorisca nell'utente la costruzione di una mappa cognitiva più completa, capace di supportare successivamente task che richiedono l'orientamento nell'ambiente, e che questa conoscenza sia poi traferibile nello svolgimento dei task in modalità desktop.

Al contrario, l'esperienza desktop non permette la costruzione di una conoscenza accurata dell'ambiente anche per i limiti che abbiamo visto sopra (FOV limitato ecc), per cui la

prima esperienza desktop non è stata in grado di influenzare fortemente gli utenti nello svolgimento degli stessi task nel cave.

Gli stessi utenti nel passaggio dall'esperienza su PC a quella nel CAVE hanno dichiarato di aver compreso meglio la struttura dell'ambiente, in particolare la collocazione spaziale delle varie informazioni. Gli utenti che invece sono passati dalla modalità immersiva a quella desktop non ritengono di aver ottenuto una migliore comprensione, a parte alcuni che nella modalità immersiva non avevano notato la presenza del menu centrale con i link verso tutte le tematiche, maggiormente visibile invece in modalità desktop.

4.3.8 Soddisfazione dell'utente

Per valutare la soddisfazione dell'utente riguardo gli information landscapes e il sistema di visualizzazione relativo, sono state preparate domande apposite nel questionario relative a l'opinione generale dell'utente, l'opinione sulla modalità di visualizzazione, e l'opinione sulla tecnica di interazione.

Opinione generale

La prima impressione riguardo il sistema all'inizio del test è stata complessivamente più positiva per quanto riguarda la modalità immersiva rispetto alla modalità desktop (Tabella 33). Nella prima modalità infatti non si evidenziano opinioni negative, presenti invece per un totale del 25% degli utenti nella modalità desktop. Si può inoltre osservare come in genere le opinioni degli utenti esperti siano in genere meno positive rispetto agli utenti inesperti, forse perché i primi sono maggiormente influenzati nel giudizio dal fatto di aver sperimentato altri sistemi a loro parere migliori, o comunque perché sono convinti della possibilità di fare meglio.

Modalità desktop

	<i>tutti utenti</i>	<i>esperti</i>	<i>inesperti</i>
molto negativa	0%	0%	0%
negativa	8%	17%	0%
AbbNegativa	17%	17%	17%
indifferenza	33%	33%	33%
abbPositiva	17%	17%	17%
positiva	25%	17%	33%
Molto positiva	0%	0%	0%

Modalità immersiva

	<i>tutti utenti</i>	<i>esperti</i>	<i>inesperti</i>
molto negativa	0%	0	0%
negativa	0%	0%	0%
abbnegativa	0%	0%	0%
indifferenza	33%	67%	0%
abb positiva	17%	17%	17%
positiva	50%	17%	83%
molto positiva	0%	0%	0%

Tabella 33 Risultati domanda: "Qual è stata la tua impressione generale riguardo il sistema all'inizio del test?"

Dopo l'utilizzo del sistema durante il test, l'opinione è cambiata (Tabella 34). Nel caso desktop l'opinione si fa più negativa, forse a causa dell'emergere di diversi problemi di usabilità; nel caso cave, si nota una diminuzione dell'opinione "positiva" a favore di quella "abbastanza positiva", ma anche una diminuzione della percentuale di utenti indifferenti a favore di un'opinione positiva.

	<i>tutti utenti</i>	<i>esperti</i>	<i>inesperti</i>
molto negativa	0%	0%	0%
negativa	42%	33%	50%
abbNegativa	25%	33%	17%
indifferenza	0%	0%	0%
abbPositiva	25%	17%	33%
positiva	8%	17%	0%
molto positiva	0%	0%	0%

	<i>tutti utenti</i>	<i>esperti</i>	<i>inesperti</i>
molto negativa	0%	0%	0%
negativa	0%	0%	0%
abb negativa	0%	0%	0%
indifferenza	8%	17%	0%
abb positiva	50%	50%	50%
positiva	33%	33%	33%
molto positiva	8%	0%	17%

Tabella 34 risultati domanda: "Qual è stata la tua impressione generale riguardo il sistema dopo che lo hai utilizzato?"

Chi ha cambiato opinione negativamente riguardo la modalità desktop giustifica affermando che la modalità di interazione è particolarmente scomoda e sicuramente da migliorare. In particolare sono infastiditi dalla eccessiva sensibilità del mouse.

Nella modalità immersiva, chi ha cambiato opinione negativamente, afferma che l'idea degli IL è potenzialmente buona, ma in pratica il sistema non fornisce una buona navigazione, né fornisce strumenti per facilitarla; alcuni lamentano un'eccessiva velocità del joystick. Chi ha cambiato l'opinione positivamente afferma che il sistema era meno scomodo di quanto si aspettasse, che il sistema gli ha consentito una piena interazione con l'ambiente virtuale, e ha favorito la comprensione della struttura della scena.

Dopo l'esperienza avuta, buona parte degli utenti desktop (circa il 50%) afferma di non voler più avere niente a che fare con questo sistema, mentre nella modalità immersiva la maggior parte degli utenti userebbe di nuovo gli ILs (Tabella 35).

Modalità desktop

	<i>tutti utenti</i>	<i>esperti</i>	<i>inesperti</i>
si	25%	17%	33%
no	50%	50%	50%
forse	25%	33%	17%

Modalità immersiva

	<i>tutti utenti</i>	<i>esperti</i>	<i>inesperti</i>
si	67%	50%	83%
no	8%	0%	17%
forse	25%	50%	0%

Tabella 35 Risultati domanda: "Utilizzeresti di nuovo questo sistema?"

Circa il 50% degli utenti che hanno giudicato il sistema desktop attualmente inefficace, ritiene che comunque esso abbia potenzialità interessanti qualora in futuro venga migliorato; nella modalità immersiva gli utenti risultano essere il 100% di quelli che hanno avuto un giudizio negativo (Tabella 36).

Modalità desktop

	<i>tutti utenti</i>
si	50%
no	13%
forse	38%

Modalità immersiva

	<i>tutti utenti</i>
si	100%
no	0%
forse	0%

Tabella 36 Risultati domanda: "Se nell'attuale implementazione non ritieni questo strumento efficace, ritieni che abbia comunque potenzialità interessanti qualora venga migliorato?"

Gli aspetti che a parere degli utenti devono essere migliorati sia nella modalità desktop che immersiva, riguardano soprattutto l'interazione e il supporto all'orientamento (Figura 56).

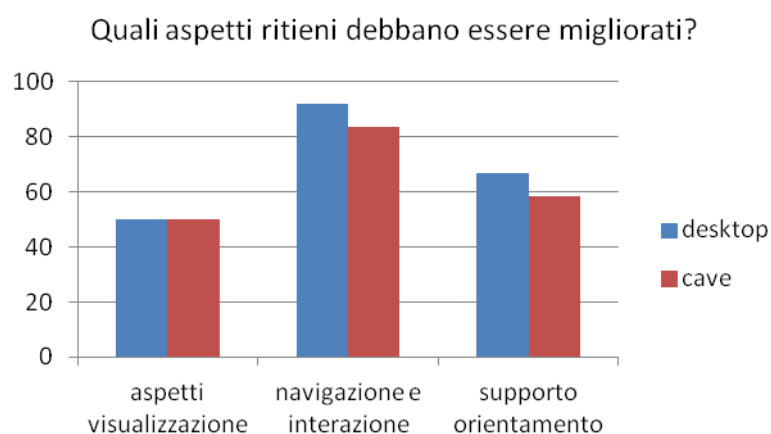


Figura 56 Aspetti Che gli utenti ritengono debbano essere migliorati (modalità desktop e immersiva)

Gli ambiti di applicazione dei due sistemi di visualizzazione e interazione con gli ILs secondo gli utenti, sono riportati nel grafico seguente (Figura 57):

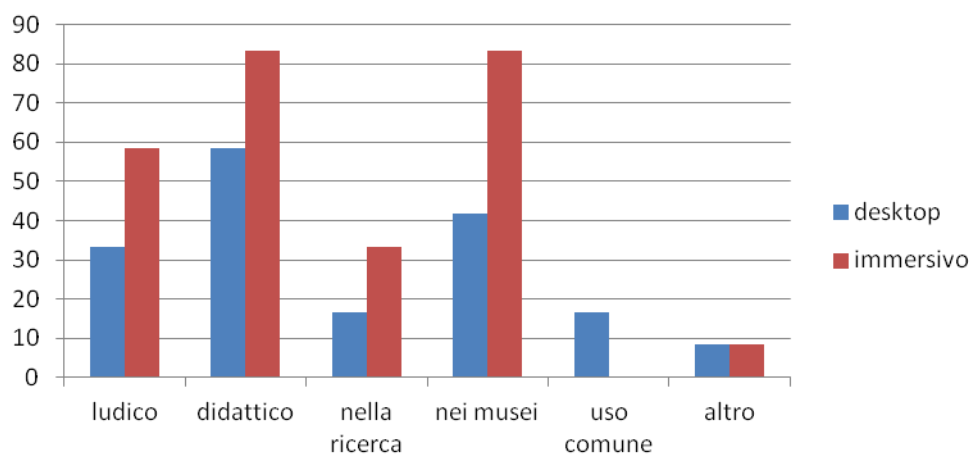


Figura 57 Ambiti di applicazione degli IL secondo gli utenti (nelle due modalità di fruizione, desktop e immersiva)

Opinioni sulla tecnica di interazione

All'efficacia della tecnica di interazione con il mouse, gli utenti hanno dato globalmente il voto di 3/5 mentre a quella con il joystick è stato dato il voto di 4/5 (Figura 37).

MOUSE

	media voti
esperti	2,5
inesperti	3,5
tutti	3,0

JOYSTICK

	media voti
esperti	3,8
inesperti	4,2
tutti	4,0

Tabella 37 Risultati domanda: " Una volta imparata, quanto ritieni efficace la tecnica di interazione con il mouse/joystick?"

Agli utenti è stato chiesto di indicare uno o più aggettivi che meglio identificano la tecnica di interazione con il mouse. Le espressioni che meglio identificano l'interazione con il mouse sono "poco intuitivo" e "dispendioso", mentre quelle utilizzate per il joystick sono "intuitivo", "efficace", "semplice" (Figura 58).

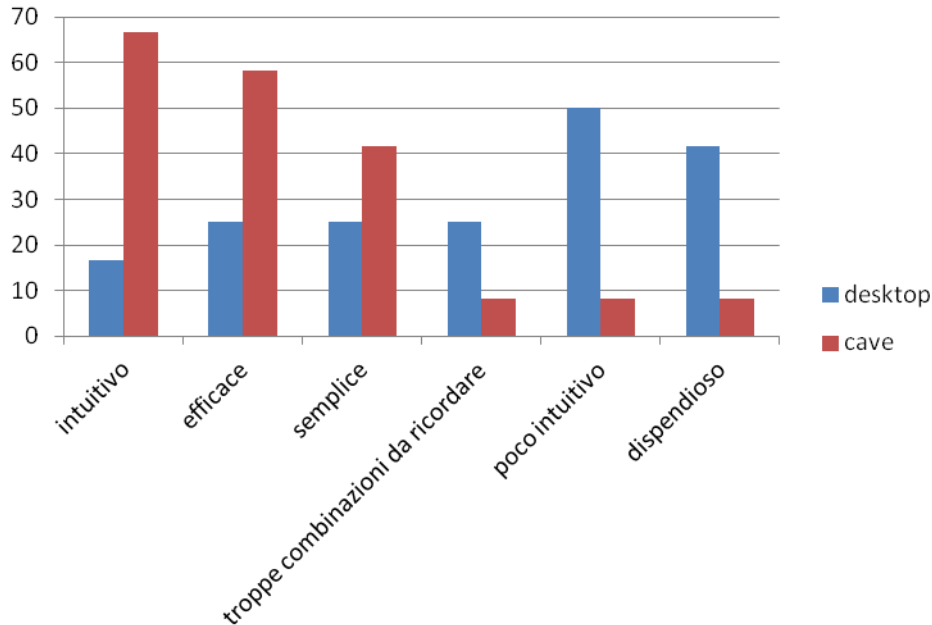


Figura 58 Opinioni sulla tecnica di interazione (confronto tra modalità desktop e immersiva)

Per quanto riguarda la tecnica di interazione con il mouse, gli utenti indicano i seguenti possibili miglioramenti:

- utilizzare il click sinistro per attivare i collegamenti e il destro per la modalità di navigazione automatica
- sostituire la rotazione con il pulsante sinistro del mouse con un free look
- ridurre la sensibilità nella rotazione
- risolvere il problema della cattiva integrazione tra navigazione libera e navigazione automatica: i link e la modalità di navigazione automatica al momento si attivano su un evento di mouse up anche quando il mouse down è iniziato fuori dall'elemento.
- ripensare il modo in cui i tasti sono assegnati alle varie modalità

Per quanto riguarda la tecnica di interazione con il joystick, gli utenti indicano i seguenti miglioramenti:

- inserire la possibilità di regolare la velocità negli spostamenti.
- velocizzare gli spostamenti automatici
- per gli spostamenti in alto e in basso, utilizzare una coppia di tasti anziché una combinazione tasto più leva oppure la levetta di destra.

- trovare una soluzione diversa per la selezione dei link nel caso di elementi di piccola dimensione oppure assegnare i link solo ad oggetti di dimensione medio-grande.

Complessivamente il voto dato dagli utenti alla tecnica di interazione è di 3/5 per la modalità desktop, mentre per la modalità immersiva è di 4/5.

Opinioni sulla modalità di visualizzazione

I vantaggi che i due sistemi - desktop e immersivo - di visualizzazione di ILs hanno apportato rispetto alle modalità più tradizionali sono riassunti nel seguente grafico (Figura 59) che riporta per ciascun possibile vantaggio, la percentuale di utenti:

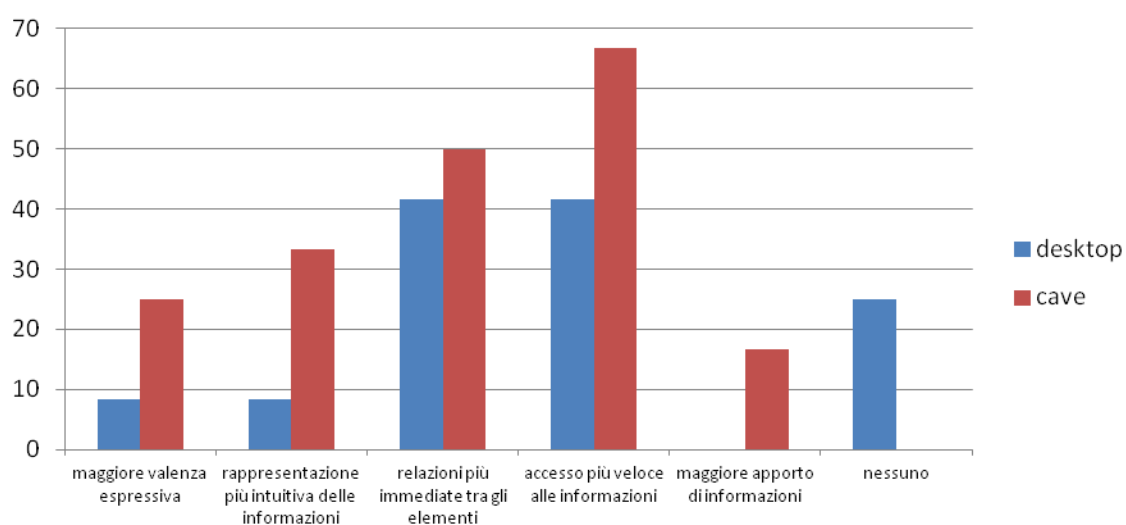


Figura 59 Vantaggi degli IL nelle due modalità di fruizione

Ciò che colpisce di più, è il fatto che una significativa percentuale di utenti ritiene che il sistema desktop non presenti alcun vantaggio.

Il sistema desktop a parere degli utenti ha infatti lo svantaggio di presentare una minore valenza espressiva, di proporre una rappresentazione meno intuitiva delle informazioni, relazioni meno immediate tra gli elementi e di fornire un minor apporto di informazioni rispetto al sistema immersivo (Figura 60).

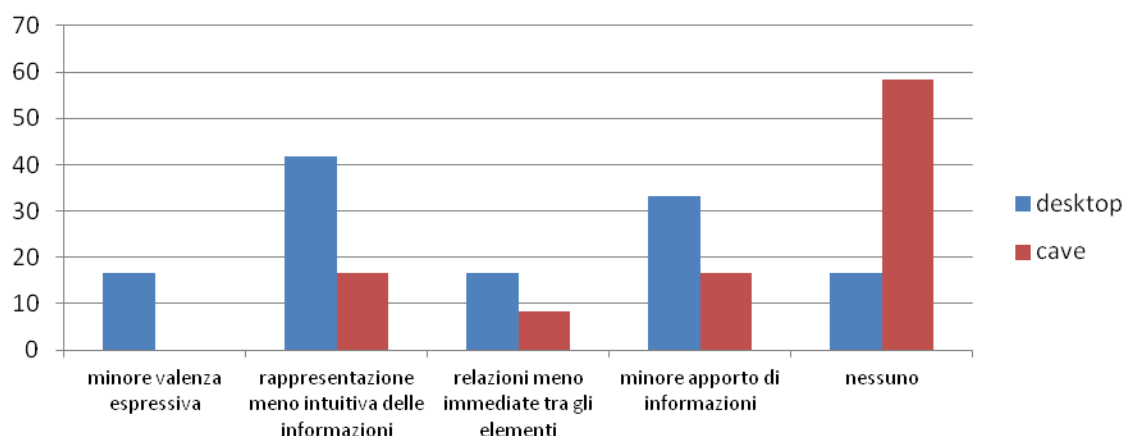


Figura 60 Svantaggi degli IL nelle due modalità di fruizione

Per quanto riguarda la lettura dei testi (Tabella 38), in entrambe le modalità gli utenti hanno avuto molte o alcune difficoltà, ma gli utenti che non hanno avuto nessuna difficoltà sono superiori nella modalità immersiva (58%) rispetto alla modalità desktop (33%). Il risultato un po' stupisce perché ci saremmo aspettati una maggiore difficoltà nel leggere testi in modalità stereo.

Modalità desktop

	tutti utenti
alcune	50%
nessuna	33%
molte	17%

Modalità immersiva

	tutti utenti
alcune	33%
nessuna	58%
molte	8%

Tabella 38 Risultati domanda: "Hai trovato difficoltà nella lettura dei testi in 3 dimensioni?"

La soluzione di utilizzare un quadratino per indicare la presenza di un link è ritenuta dal 67% degli utenti efficace e solo dal 33% non efficace.

Solo 3 utenti su 12 (25%) hanno manifestato disagio durante la visione dell'applicazione in modalità immersiva. I maggiori fastidi segnalati sono: stanchezza alla vista, nausea e mal di testa.

Agli utenti è stato richiesto di riassumere con un voto la loro opinione riguardo la modalità di visualizzazione. I risultati sono riportati nelle tabelle seguenti (Tabella 39):

Modalità desktop

	<i>media voti</i>
esperti	2,7
inesperti	2,8
tutti	2,8

Modalità immersiva

	<i>media voti</i>
esperti	3,2
inesperti	3,7
tutti	3,4

Tabella 39 Risultati domanda: "Riassumi con un voto da 1 a 5 la tua opinione riguardo la modalità di visualizzazione"

Giudizio complessivo degli utenti

Il seguente grafico (Figura 61) riassume il giudizio complessivo degli utenti in merito alla modalità di visualizzazione, la tecnica di interazione, i tempi di apprendimento della tecnica di interazione, e riporta inoltre il voto complessivo, dato dalla media dei tre aspetti.

Mentre il giudizio complessivo sulla fruizione immersiva con joystick è indice di una usabilità soddisfacente, la fruizione desktop necessita di essere migliorata affinché l'utente gradisca veramente l'utilizzo di questi sistemi. Il voto complessivo ottenuto (3/5) indica ancora un certo scetticismo verso gli information landscape su PC, anche se questo è molto diverso da un rifiuto vero e proprio.

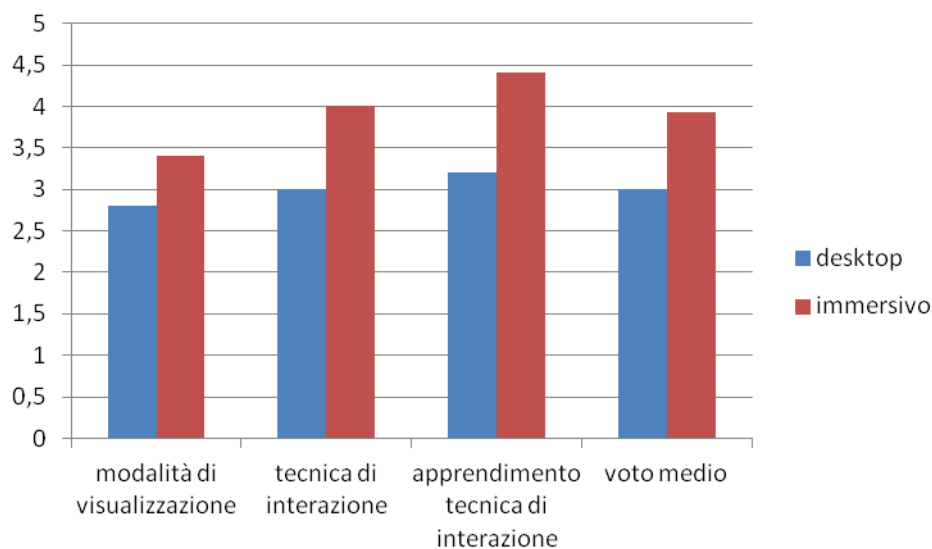


Figura 61 voti dati dagli utenti a vari aspetti degli IL nelle due modalità di fruizione

4.4 Conclusioni sul test

Date le risorse limitate - tempi ridotti, scarsa disponibilità di utenti, scarsa esperienza nell'ambito dell'usabilità - l'esperimento che si è preparato è da considerare come un primo passo. Ciò nonostante esso ha permesso di ottenere una serie di risultati interessanti, da cui partire eventualmente per migliorare l'attuale visualizzatore per IL e per progettare test più significativi e più mirati. Qui di seguito elenchiamo i risultati che abbiamo ottenuto, i maggiori problemi di usabilità rilevati, e si riflette su alcune possibili soluzioni. Di alcune di queste è stata effettuata l'implementazione, altre saranno da prendere in considerazione per sviluppi futuri.

4.4.1 Riepilogo delle osservazioni fatte durante il test

In questo paragrafo si riepilogano le osservazioni che emergono dall'analisi dei risultati del test.

1. Il tempo di apprendimento della tecnica di interazione del mouse e del joystick sono ridotti e molto simili tra loro

Gli utenti hanno mediamente impiegato tempi simili nell'apprendimento della tecnica di interazione, anche se leggermente superiori nel caso del joystick. I tempi sono comunque molto bassi e teoricamente accettabili per un applicazione di IL.

2. Gli utenti hanno percepito come più faticoso l'apprendimento del mouse

Nonostante i tempi di apprendimento siano più o meno gli stessi, la maggior parte degli utenti ha percepito come faticoso l'apprendimento della tecnica di interazione con il mouse

3. La tecnica di interazione con il joystick permette di eseguire in minor tempo task di navigazione

Gli utenti, nella maggior parte dei task di navigazione, hanno impiegato meno tempo con il joystick che con il mouse.

4. Gli utenti percepiscono come troppo elevati i tempi impiegati nell'accesso all'informazione utilizzando il mouse

Anche se i tempi di accesso all'informazione con il mouse sono superiori ma non di molto, gli utenti li percepiscono come troppo elevati.

5. La tecnica di interazione con il joystick è preferita dagli utenti

Gli utenti hanno trovato più semplice ed efficace l' utilizzo del joystick per effettuare task di navigazione. Anche dal punto di vista della soddisfazione, l'utente ha preferito fortemente l'interazione con il joystick rispetto a quella con il mouse.

6. La modalità di visualizzazione immersiva sembrerebbe favorire l'orientamento.

La modalità di visualizzazione immersiva sembrerebbe favorire l'apprendimento della struttura dell'ambiente e la costruzione di una mappa cognitiva da parte dell'utente che poi può essere sfruttata nell'esecuzione in modo efficiente di compiti che richiedono capacità di orientamento.

7. La modalità di visualizzazione immersiva è stata preferita dagli utenti

La modalità immersiva è stata preferita rispetto a quella desktop, e alcuni sostengono che non effettuerebbero volentieri una nuova esperienza con gli ILs in ambiente desktop.

Tali risultati permettono di affermare che forse la modalità di fruizione migliore per gli ILs è quella immersiva con il joystick, mentre quella desktop si potrebbe considerare solamente un compromesso che permette a chi non ha maggiori possibilità di fruire di questi ambienti, ma senza poter beneficiare dei vantaggi che gli ILs comportano.

Tuttavia, è proprio sulla modalità desktop che occorre investire energie se vogliamo che questi sistemi possano maggiormente diffondersi in futuro, dato che tutti possiedono un pc e un mouse mentre solo in pochi possono permettersi l'esperienza all'interno di un CAVE.

E' probabile che eventuali miglioramenti della tecnica di interazione e visualizzazione desktop possa consentire di superare alcune di queste problematiche.

Sia per mezzo dell'osservazione, sia analizzando il feedback da parte degli utenti è stato possibile individuare alcuni dei maggiori problemi di usabilità e ipotizzare delle possibili soluzioni.

4.4.2 Riepilogo dei maggiori problemi di usabilità in modalità desktop

I maggiori problemi individuati nella tecnica di interazione desktop sono:

- troppe combinazioni di tasti da ricordare

- le due modalità di navigazione (navigazione automatica e libera) non erano ben integrate tra loro: l'utente attivava senza volerlo la navigazione automatica su un oggetto durante la navigazione libera.

- eccessiva sensibilità del mouse

I maggiori problemi nella modalità di visualizzazione desktop sono correlate allo scarso FOV che lo schermo fornisce e che determina problemi nella percezione degli spazi.

4.4.3 Riepilogo dei problemi di usabilità in modalità immersiva

Sebbene la modalità immersiva presenti decisamente meno problemi rispetto a quella desktop, ci sono sicuramente aspetti che necessitano di essere presi in considerazione. I principali sono:

- il problema della selezione dei link, troppo difficile e dispendiosa al momento;
- alcune combinazioni di tasti scomode come quella per andare in alto e in basso

4.4.4 Possibili soluzioni ai problemi di usabilità nella modalità desktop

Nella modalità di fruizione desktop il problema delle troppe combinazioni di tasti da ricordare nell'utilizzo del mouse, non può essere superato se non utilizzando un altro dispositivo per l'interazione. Il mouse infatti non nasce per l'interazione con ambienti tridimensionali (pochi bottoni, 2 gradi di libertà) per cui l'unico modo di utilizzarlo in un contesto 3D è quello di inserire combinazioni di tasti. Visto la soddisfazione che gli utenti hanno mostrato nei confronti del joystick potremmo pensare di consigliare il joystick anche nella fruizione desktop degli information landscapes. Il punto è che non tutti possiedono un joystick mentre più o meno chiunque possiede il mouse. Diventa necessario quindi, quanto meno, migliorare l'attuale tecnica di interazione con il mouse.

Un problema che è stato risolto è quello dell'integrazione tra tecnica di navigazione libera e automatica che portava l'utente a commettere errori.

Un altro aspetto che è stato migliorato è quello dell'eccessiva sensibilità del mouse nelle rotazioni. Adesso dovrebbe essere più semplice controllare i movimenti del mouse.

Il problema dello scarso FOV che lo schermo fornisce può essere superato, forse, utilizzando una mappa dall'alto dell'ambiente che fornisca una consapevolezza all'utente della sua collocazione e una visione di massima della struttura dell'ambiente.

Un primo prototipo di mappa è stato sviluppato nell'ambito di questa tesi. L'obiettivo che ci siamo posti nella progettazione della mappa è quello di facilitare l'utente nella comprensione della propria collocazione nel Paesaggio di Informazione, e di avere una visione di massima della struttura dell'ambiente così da poter capire dove e come raggiungere altre zone del paesaggio di informazione.

Un primo problema da affrontare è stato quello di decidere quali oggetti visualizzare. Un Paesaggio di Informazione ha infatti una struttura gerarchica costituita da parchi, siti, pagine, livelli di dettaglio e oggetti informativi. Nel viewport principale solo gli oggetti foglie sono mostrati, mentre gli altri servono soprattutto in fase di design ad agevolare il processo di costruzione del paesaggio e fornire una strutturazione logica significativa dei contenuti. Scegliere di visualizzare le foglie avrebbe portato sicuramente ad avere una mappa troppo caotica, dato che tali contenuti solitamente sono molti. Di default, si è scelto quindi di visualizzare entità di più alto livello gerarchico e cioè le pagine. Comunque l'utente, all'occorrenza deve poter richiedere la visualizzazione di un altro livello gerarchico. Le varie entità vengono visualizzate come punti colorati. Un triangolo bianco al centro, invece, indica la camera.

Un secondo problema è dato dal fatto che un paesaggio di informazione non vincola gli oggetti a stare su un piano. Esse sono liberamente collocabili nello spazio, anche sull'asse verticale. Una visione dall'alto di un tale ambiente porterebbe a non avere una percezione di come siano distribuiti gli elementi sull'asse verticale. Abbiamo pensato dunque che una soluzione potesse essere quella di utilizzare colori diversi per veicolare questa informazione. In particolare in verde vengono disegnati gli oggetti che si trovano ad altezza di camera, in rosso gli oggetti che stanno più in alto e in blu quelli che stanno più in basso.

In una mappa così progettata (Figura 62), l'utente ha istante per istante un'idea della sua posizione nell'ambiente, di quali siano gli oggetti vicini, e anche se ci siano oggetti in alto o in basso.

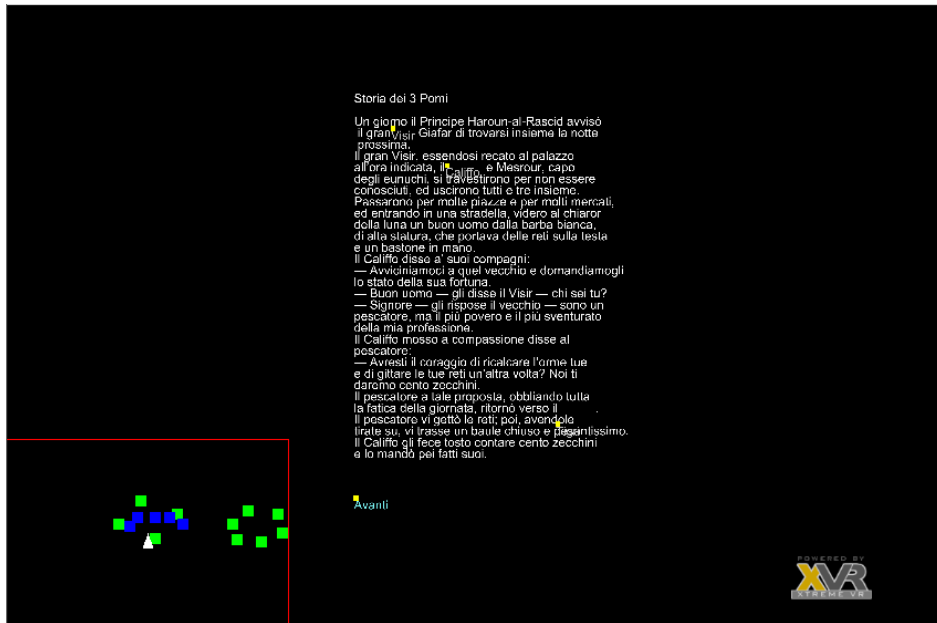


Figura 62 Mappa per information landscapes

La visualizzazione della mappa può essere richiesta facendo clic sulla select box *Show ILMap*. Per controllare il livello gerarchico delle entità che vogliamo siano visualizzate nella mappa è presente una interfaccia posta al di sotto della 3d view (Figura 63).

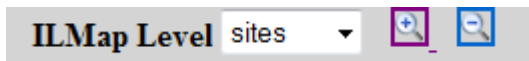


Figura 63 Interfaccia per il controllo della mappa

Essa consente inoltre di controllare la porzione di ambiente che vogliamo sia visualizzata nella mappa, tramite controlli a forma di lente di ingrandimento che applicano una sorta di zoom (zoom in o zoom out) nella mappa.

Il comando inviato dalla pagina web per la modifica del livello gerarchico visualizzato è:

```
['ilmaplevel', <level>]
```

dove <level> può valere "ILSite", "ILPage" o "ILObj".

I comandi per effettuare lo zoom in e lo zoom out sulla mappa sono rispettivamente:

```
['ilmapZoomIn']
```

```
['ilmapZoomOut']
```

Dal punto di vista realizzativo, la mappa viene implementata dalla classe `Map`⁴²:

```
class Map
{
    SetContent(content);
    Draw();

    var SceneRangeWidth;
    var SceneRangeHeight;
    var MapPositionX;
    var MapPositionY;
    var MapWidth;
    var MapHeight;
    var RangeCamera;
};
```

Posizione e dimensione della mappa all'interno della finestra sono regolabili rispettivamente tramite le variabili di istanza *MapPositionX* / *MapPositionY* e *MapWidth* / *MapHeight*. La posizione deve essere espressa nel sistema di riferimento della finestra⁴³ normalizzato all'intervallo [0,1]. Analogamente, le dimensioni della mappa vengono espresse come frazione delle dimensioni della finestra, quindi anch'esse devono appartenere all'intervallo [0,1]. Questo approccio ha il vantaggio di consentire un ridimensionamento ed un riposizionamento automatico della mappa al variare delle dimensioni della finestra.

Gli elementi da visualizzare all'interno della mappa vengono specificati per mezzo di una lista⁴⁴ contenente la loro posizione espressa nel sistema di riferimento globale. Tale lista deve essere assegnata all'oggetto `Map` tramite il metodo *SetContent*. La porzione di scena tridimensionale da visualizzare all'interno della mappa (centrata sulla posizione corrente della camera) è invece regolabile tramite le variabili di istanza *SceneRangeWidth* e *SceneRangeHeight*.

Una volta configurati questi parametri⁴⁵, è possibile disegnare a schermo la mappa invocando ad ogni frame il metodo *Draw*. Questo metodo trasforma i punti forniti col metodo *SetContent* nel sistema di riferimento locale alla mappa e disegna i relativi elementi con un diverso colore in funzione della loro “altezza” (coordinata y) rispetto alla

⁴² file `ilmap.s3d`

⁴³ L'origine si trova nell'angolo in basso a sinistra della finestra

⁴⁴ Più precisamente un array nel linguaggio `s3d`

⁴⁵ La configurazione della mappa avviene tipicamente nella fase di inizializzazione dell'applicazione ma, se necessario, è comunque possibile modificare il valore dei vari parametri a run-time.

camera, in modo da dare all'utente una migliore indicazione visiva sulla loro collocazione spaziale. In particolare:

- Vengono disegnati in verde gli elementi con “altezza” paragonabile a quella della camera, ovvero quegli elementi la cui coordinata y è compresa in un dato intervallo centrato sulla coordinata y della camera.
- Vengono disegnati in rosso gli elementi con altezza maggiore di quella della camera (oltre un certo valore di soglia).
- Vengono disegnati in blu gli elementi con altezza inferiore a quella della camera (al di sotto di un certo valore di soglia).

Il range usato per determinare la colorazione degli elementi è regolabile attraverso la variabile di istanza *RangeCamera*⁴⁶.

La mappa è stata sviluppata come componente separato in modo da renderla indipendente dalle strutture dati utilizzate dal visualizzatore di Information Landscapes, rendendolo di fatto utilizzabile anche in altri contesti.

La lista dei punti da visualizzare viene aggiornata ogni qualvolta vi è un cambiamento nella scena (creazione, modifica, cancellazione di un elemento), quando l'utente richiede la visualizzazione di entità di diverso livello gerarchico oppure quando si richiede il caricamento di un nuovo paesaggio di informazione.

4.4.5 Possibili soluzioni ai problemi di usabilità nella modalità immersiva

Una possibile soluzione (ma che ha dei limiti) per la selezione dei link è quella di consentire di scorrere la selezione degli oggetti visibili sulla schermata principale tramite la pressione di un tasto su joystick. Lo svantaggio è quando gli oggetti sullo schermo sono tanti, problema in parte risolvibile inserendo un raggio oltre il quale l'oggetto non viene considerato come selezionabile.

Nell'ambito di questa tesi si è implementata la funzionalità descritta, ma essa almeno per il momento non si integra correttamente nell'architettura esistente. Tale soluzione infatti

⁴⁶ L'intervallo all'interno del quale un elemento viene considerato “alla stessa altezza” della camera è quindi $[camera.y - RangeCamera, camera.y + RangeCamera]$.

richiede la conoscenza di quali elementi sono visibili nella scena, e per comprendere questo occorre effettuare un test di culling al momento non implementato per la visualizzazione all'interno del cave.

Un'altra possibilità potrebbe essere quella di programmare una wand, un dispositivo di puntamento (una sorta di telecomando) pensato appositamente per la realtà virtuale. Nell'ambito di questa tesi non è stato possibile considerare tale soluzione in quanto per il funzionamento è necessario che nel cave venga installato un sistema di tracciamento della posizione e orientazione, al momento non presente.

Tra i comandi da cambiare vi è sicuramente la combinazione di tasti per andare in alto o in basso, che potrà essere sostituita con una delle soluzioni proposte dagli utenti: uso della levetta di sinistra, uso di due tasti per andare in alto e in basso.

La modalità di visualizzazione immersiva sembra non presentare grandi problematiche per l'orientamento, ma è tuttavia possibile pensare all'implementazione di mappe specifiche per l'utilizzo all'interno del cave. Il tal caso, un problema da risolvere sarebbe quello di decidere in che modo e in che punto del cave visualizzare la mappa.

4.4.6 Possibili linee guida per la progettazione di IL

L'osservazione del comportamento degli utenti e il loro feedback ci ha permesso di individuare alcune linee guida per la progettazione di un Paesaggio di Informazione che sia usabile oltre a quelle già segnalate in (Ruffaldi et al., 2008).

Una prima indicazione è quella di inserire testi brevi e aventi una struttura semplice all'interno di un paesaggio di Informazione. La lettura di testi in 3D è più difficoltosa, per cui se i testi sono troppo lunghi e complessi l'utente finirà per non leggerli.

Alcuni utenti hanno lamentato l'assenza di significativi punti di riferimento. Durante la progettazione bisognerà quindi dare molta importanza nella creazione di alcuni elementi facilmente memorizzabili e riconoscibili, che l'utente possa utilizzare come punti di riferimento.

E' importante inoltre rendere immediatamente riconoscibili gli elementi più importanti della struttura dell'ambiente. In CubiSpace i cubi non sono immediatamente associabili ad una tematica artistica. Dato un cubo siamo in grado di dire la tematica associata solo dopo averlo raggiunto e averne visualizzato il contenuto, oppure vedendo quale delle tematiche

presenti nel menu centrale è collegata tramite link a quello specifico cubo. Un modo più semplice per riconoscere la tematica associata al cubo poteva essere quella di inserire una scritta grande sopra ogni cubo.

Nel caso in cui la selezione dei link avvenga tramite joystick secondo la modalità descritta più volte nel corso di questa tesi, occorrerà prestare attenzione in fase di design alla grandezza degli elementi su cui applicare un collegamento. Non siamo ancora in grado di dire la grandezza minima, ma possiamo escludere sicuramente i link su parole o singole frasi.

Inoltre, attualmente i link sono indicati tramite un quadratino giallo posto in alto a sinistra sull'elemento source. Nel caso in cui due elementi siano vicini l'utente ha mostrato difficoltà a stabilire quale fosse l'oggetto con link. Un esempio di questa situazione si ha in cubispace nel menu principale (Figura 64): i link sono sulle parole eppure gli utenti continuavano a fare click sulle immagini e si lamentavano che il link non funzionava. È quindi importante evitare di inserire link su un oggetto troppo vicino ad un altro: l'oggetto con link deve essere chiaramente distinguibile da tutti gli altri.



Figura 64 menu principale in cubispace

Conclusioni e sviluppi futuri

Il lavoro di tesi si è sviluppato a partire dalla constatazione che, nonostante le potenzialità interessanti degli Information Landscapes, nella realtà tali sistemi di visualizzazione tridimensionale delle informazioni non sono stati ampiamente adottati.

Si è ipotizzato che una prima causa potesse risiedere nel fatto che non fossero stati sufficientemente diffusi sistemi che permettessero a chiunque di creare i propri Paesaggi di Informazione e di condividerli, mentre una seconda causa è probabilmente legata alla scarsa attenzione prestata all'aspetto dell'usabilità dei software che permettono la navigazione e interazione con queste applicazioni di realtà virtuale. Ci siamo perciò posti l'obiettivo di lavorare nella direzione di una soluzione a questi due aspetti.

In primo luogo, a partire dal visualizzatore XVR di Information Landscapes sviluppato presso il Laboratorio PERCRO (Scuola Superiore S. Anna di Pisa), è stato progettato e implementato il prototipo di uno strumento di editing che fornisce all'utente interfacce per la creazione di nuovi paesaggi o la modifica di quelli esistenti. In aggiunta è stata introdotta una nuova funzionalità, quella di ricerca testuale, che dovrebbe agevolare l'utente nella ricerca dell'informazione di interesse, soprattutto nel caso di ambienti complessi.

È stato inoltre effettuato uno studio di usabilità nell'interazione con i Paesaggi di Informazione per mezzo di test con utenti, allo scopo di valutare l'impatto che questa nuova metodologia di divulgazione delle informazioni ha sulle persone, e di permettere l'identificazione delle maggiori problematiche.

Il test ci ha permesso di confermare quanto ipotizzato e cioè il fatto che uno dei motivi dell'insuccesso degli information landscapes sia da imputare alla loro usabilità non pienamente soddisfacente non tanto nella modalità immersiva quanto in quella desktop. E' proprio sulla modalità desktop che occorre investire energie se vogliamo che questi sistemi possano maggiormente diffondersi in futuro, dato che tutti possiedono un pc e un mouse mentre solo in pochi possono permettersi l'esperienza all'interno di un CAVE. Partendo dai problemi identificati nel test (legati soprattutto all'interazione con il mouse, e allo scarso campo di vista nella visualizzazione) si sono ipotizzate alcune soluzioni e alcune sono state implementate. Particolarmente interessante è lo sviluppo di una mappa

per IL, la quale si pone l'obiettivo di agevolare l'orientamento dell'utente all'interno di un paesaggio.

Sviluppi Futuri

Per quanto riguarda lo strumento di editing, ci sono sicuramente aspetti che possono essere migliorati. In particolare l'editor dovrà essere testato in casi reali allo scopo di valutarne l'usabilità e apportare eventuali modifiche. Anche l'estetica dell'interfaccia può essere sicuramente migliorata. Per quanto riguarda la mappa per IL un possibile miglioramento potrebbe essere quello di determinare in modo accurato (piuttosto che utilizzando un valore di soglia) quali elementi si trovano ad altezza di camera utilizzando opportuni calcoli matematici.

In futuro potrà essere implementato un modo per regolare l'accesso al sistema e la condivisione dei paesaggi. Al momento infatti il sistema è totalmente pubblico: ogni utente può accedere, modificare e cancellare i paesaggi creati da chiunque. Lo sviluppo di un sistema di autenticazione permetterebbe a ciascun utente di visualizzare solo i propri paesaggi di informazione o al più quelli condivisi da altri utenti. Sarebbe interessante progettare un sistema che consenta all'utente di scegliere quali dei propri paesaggi debbano rimanere privati e quali pubblici.

Per quanto riguarda l'aspetto dell'usabilità degli IL, lo studio effettuato costituisce solamente un primo passo. In futuro potranno essere sviluppati test più mirati allo scopo di approfondire maggiormente le problematiche di usabilità rilevate. Per quanto riguarda le soluzioni presentate occorrerebbe verificare il feedback degli utenti per accertare che siano veramente risolutive. Inoltre, rimane ancora da trovare delle risposte ad alcune problematiche rimaste aperte come quella relativa alla modalità di selezione dei link in modalità immersiva.

In futuro potranno essere sviluppate nuove modalità di interazione, per esempio quella touchscreen nel caso desktop, o tramite wand nel caso immersivo. Potrebbe infine essere interessante affrontare il problema della creazione di una mappa per IL per la visualizzazione all'interno del CAVE.

APPENDICE A (Elenco dei comandi)

In questa appendice si fornisce la sintassi e la descrizione del funzionamento di ciascuno dei comandi scambiati tra la pagina web e l'applicazione XVR. I comandi sono raggruppati in "comandi che la pagina web riceve da XVR" e "comandi che la pagina web invia ad XVR".

COMANDI CHE LA PAGINA WEB RICEVE DA XVR

- ["structure", <structure>, <pathstructure>, <typestructure>]

Comando ricevuto al caricamento di un Paesaggio di Informazione e ad ogni cambiamento che avviene in esso (aggiunta o rimozione di elementi). Viene utilizzato dalla pagina web per caricare opportunamente la select box che mostra la struttura dello specifico paesaggio.

Parametri:

<structure> contiene un array di array nel formato [nome[nome_figlio[...]]] contenente quindi il nome di tutti gli elementi presenti.

<pathstructure> contiene un array di array nel formato [path[path_figlio[..]]] contenente il path di tutti gli elementi presenti

<typestructure> contiene un array di array nel formato [type[type_figlio[..]]] contenente il tipo per ogni elemento presente (es: ILWORLD, ILSITE, ILPAGE ecc)

- ["sel", <path>]

Comando ricevuto quando l'utente effettua una selezione di un elemento dalla 3dView. Necessario alla pagina web per aggiornare l'interfaccia grafica (elemento visualizzato nella select box struttura, pannelli di editing visualizzati).

Parametri:

<path> è il percorso dell'elemento selezionato (es: /sito/pagina/0/testo)

- **["text_info", <text>, , <color>, <pos>, <scale>]**

Comando ricevuto in risposta ad un comando di selezione (vedi comando "sel" inviato dalla pagina web ad XVR) di un elemento della classe ILTEXT. Serve ad aggiornare i campi dell'interfaccia di editing relativi all'elemento testuale.

Parametri:

<text> è il testo contenuto nel particolare elemento testuale

 è il tipo di font (es: Arial, Algerian)

<color> è un vettore contenente l'informazione di colore [r,g,b], dove r,g,b sono valori compresi tra 0 e 1

<pos> è un vettore [x,y] contenente la posizione dell'elemento testuale

<scale> è un vettore contenente la scala del testo

- **["image_info", <caption>, <filename> <pos>, <size>]**

Comando ricevuto in risposta ad un comando di selezione (vedi comando "sel" inviato dalla pagina web ad XVR) di un elemento della classe ILIMAGE. Serve ad aggiornare i campi dell'interfaccia di editing relativi all'elemento testuale.

Parametri:

<caption> è il testo contenuto nella didascalia della particolare immagine

<filename> è il path del file immagine sul server (es: upload/images/imm.jpg)

<color> è un vettore contenente l'informazione di colore [r,g,b], dove r,g,b sono valori compresi tra 0 e 1

<pos> è un vettore contenente la posizione dell'elemento

<size> è un vettore contenente la grandezza dell'immagine

- **["model_info", <filename>, <pos>, <scale>, <orient>]**

Comando ricevuto in risposta ad un comando di selezione(vedi comando "sel" inviato dalla pagina web ad XVR) di un elemento della classe ILMODEL. Serve ad aggiornare i campi dell'interfaccia di editing relativi al modello 3D.

Parametri:

<filename> è il path del modello 3D sul server (es: upload/3d_models/mod.aam)

<pos> è un vettore contenente la posizione dell'elemento

<scale> è un vettore contenente la scala dell'elemento

<orient> è un vettore (angolo, asse) contenente l'orientazione dell'elemento

- **["page_info", <pos>, <orient>]**

Comando ricevuto in risposta ad un comando di selezione (vedi comando "sel" inviato dalla pagina web ad XVR) di un elemento della classe ILPAGE. Serve ad aggiornare i campi dell'interfaccia di editing relativi alla pagina.

Parametri:

<pos> è un vettore contenente la posizione dell'elemento

<orient> è un vettore (angolo, asse) contenente l'orientazione dell'elemento

- **["site_info", <pos>, <orient>]**

Comando ricevuto in risposta ad un comando di selezione (vedi comando "sel" inviato dalla pagina web ad XVR) di un elemento della classe ILSITE. Serve ad aggiornare i campi dell'interfaccia di editing relativi alla pagina.

Parametri:

<pos> è un vettore contenente la posizione dell'elemento

<orient> è un vettore (angolo, asse) contenente l'orientazione dell'elemento

- **["world_info", <startPos>, <backgroundColor>]**

Comando ricevuto in risposta ad un comando di selezione (vedi comando "sel" inviato dalla pagina web ad XVR) di un elemento della classe ILWORLD. Serve ad aggiornare i campi dell'interfaccia di editing relativi alla radice.

Parametri:

<startPos> è un vettore contenente la posizione iniziale della camera al momento dell'apertura del paesaggio

<backgroundColor> è un vettore contenente l'informazione di colore [r,g,b], dove r,g,b sono valori compresi tra 0 e 1

- ['save', <ilxb>, <fileuploaded>]

Comando ricevuto in risposta ad un comando 'save' (inviato dalla pagina web ad XVR). Invia alla pagina web le informazioni necessarie al salvataggio dell'IL corrente. In particolare i parametri:

<ilxb> è una stringa esadecimale di descrizione dell' IL corrente

<fileUploaded> è una stringa contenente la lista dei nuovi file caricati in fase di creazione dell'IL, separati da *

- ['currentIL', <currentIL>]

Comando ricevuto dalla pagina web al momento del caricamento di un nuovo IL che comunica il path sul server dell'IL correntemente visualizzato (es: ILs/esempio.zip)

- ['pos', <pos>]

Comando ricevuto quando effettuiamo uno spostamento di un certo oggetto selezionato utilizzando il gizmo di traslazione. Comunica alla pagina web i nuovi valori di posizione per l'oggetto selezionato.

Parametri:

<pos> è un vettore contenente i valori di posizione dell'oggetto selezionato

- ['rot', <rot>]

Comando ricevuto quando effettuiamo una rotazione di un certo oggetto selezionato utilizzando il gizmo di rotazione. Comunica alla pagina web i nuovi valori di rotazione per l'oggetto selezionato.

Parametri:

<rot> è un vettore (angolo, asse) contenente i valori di rotazione dell'oggetto selezionato

- ['scale', <scale>]

Comando ricevuto quando effettuiamo una scalatura di un certo oggetto selezionato utilizzando il gizmo di scalatura. Comunica alla pagina web i nuovi valori di scalatura per l'oggetto selezionato.

Parametri:

<scale> è un vettore contenente i valori di scalatura dell'oggetto selezionato. Nel caso delle immagini il vettore contiene la dimensione dell'immagine [sizeX, sizeY]

COMANDI CHE LA PAGINA WEB INVIA ALL'APPLICAZIONE XVR

- ['showGizmos', <boolean>]

Comando per visualizzare o no i gizmo dell'elemento selezionato.

Parametri:

<boolean> se true attiva la visualizzazione del gizmo, se false la disattiva

- ['showmap', <boolean>]

Comando per attivare o disattivare la visualizzazione della mappa per IL

Parametri:

<boolean> se true attiva la visualizzazione, se false la disattiva

- ['goto', <obj_path>]

Comando per attivare la navigazione automatica verso un oggetto

Parametri:

<obj_path> è l'oggetto verso il quale vogliamo navigare

- ['sel', <path>]

Comando per comunicare all'applicazione XVR quale oggetto è stato selezionato da interfaccia web

Parametri:

<path> è l'oggetto che è stato selezionato

- ['new', <ctx>, 'ILLEVEL']

Comando usato per la creazione di un nuovo livello di dettaglio. Il parametro

<ctx> indica il path dell'oggetto del quale verrà creato il figlio.

- ['new', <ctx>, 'ILPAGE', <name>]

Comando usato per la creazione di un nuovo oggetto ILPAGE.

Parametri:

<ctx> indica il path dell'oggetto del quale verrà creato il figlio

<name> indica il nome che si vuole dare alla pagina creata

- ['new', <ctx>, 'ILSITE', <name>]

Comando usato per la creazione di un nuovo oggetto ILSITE.

Parametri:

<ctx> indica il path dell'oggetto del quale verrà creato il figlio

<name> indica il nome che si vuole dare al sito creato

- ['new', <ctx>, 'ILTEXT', <name>, <text>]

Comando usato per la creazione di un nuovo oggetto ILTEXT.

Parametri:

<ctx> indica il path dell'oggetto del quale verrà creato il figlio

<name> indica il nome che si vuole dare all'oggetto testuale

<text> è il testo che vogliamo assegnare all'oggetto testuale

- ['new', <ctx>, 'ILIMAGE', <name>, <url>]

Comando usato per la creazione di un nuovo livello oggetto ILIMAGE.

Parametri:

<ctx> indica il path dell'oggetto del quale verrà creato il figlio

<name> indica il nome che si vuole dare all'immagine

<url> indica l'url dell'immagine sul server

- ['new', <ctx>, 'ILIMODEL', <name>, <url>]

Comando usato per la creazione di un nuovo livello oggetto ILMODEL.

Parametri:

<ctx> indica il path dell'oggetto del quale verrà creato il figlio

<name> indica il nome che si vuole dare al modello 3D

<url> indica l'url del modello 3D sul server

- ['remove', <path>]

Comando usato per richiedere la cancellazione di un oggetto dal Paesaggio.

Parametri:

<path> indica il path dell'oggetto che vogliamo eliminare

- **['clear']**
Comando usato per richiedere la cancellazione dell'intero paesaggio di informazione correntemente visualizzato, ad eccezione della radice.
- **['changeGizmoType', <type>]**
Comando usato per specificare il tipo di operazione di editing che si vuole fare con i gizmo.
Il gizmo visualizzato cambierà di conseguenza.
Parametri:
<type> può valere "translate", "scale", "rotate", "uniformscale"
- **['back']**
['forward']
Comandi usati per andare avanti e indietro nella history degli oggetti visualizzati (funzionalità al momento non funzionante in modo corretto)
- **['search', <text>]**
Il comando richiede una ricerca di un determinato testo nel Paesaggio di Informazione e produce una visualizzazione grafica dei risultati.
Parametri:
<text> può essere una parola, più parole separate da spazio, o una espressione speciale.
Per esempio:
parola ricerca tutti i record contenenti *parola*
parola1 parola2 ricerca tutti i record che contengono la *parola1* e la *parola2*
*p** ricerca tutti i record contenenti parole che iniziano per *p*
parola1 OR parola2 ricerca tutti i record che contengono una parola o l'altra
parola1 - parola2 ricerca tutti i record che contengono la *parola1* ma non la *parola2*
- **['save']**
Comando con cui si richiede il salvataggio dell'IL corrente. In risposta a questo comando l'applicazione XVR invia alla pagina web un comando save contenente le informazioni per il salvataggio.
- **['loadIL', <pathIL>]**
Comando con cui si richiede di un nuovo IL presente sul server.
Parametro:
<pathIL> è il path sul server dell'IL che vogliamo caricare (es: ILs/nuovoIL.zip)
- **['singlewords', <boolean>]**
Comando con cui si specifica la modalità di ricerca testuale desiderata (e quindi di visualizzazione dei risultati).
Parametro:

<boolean> Se boolean è false la ricerca è per elementi, altrimenti è per parole.

- **['newlink']**
Comando con cui si richiede la creazione di un link tra due elementi che sono stati selezionati nella 3d view
- **['linkmodify',degree,viewpoint,time]**
Comando con cui si richiede la modifica dei parametri di un link selezionato
Parametri:
<degree> stringa testuale che esprime il grado della curva del link
<viewpoint> stringa testuale nel formato "x,y,z" che esprime la posizione della camera relativamente all'elemento destinazione del link
<time> stringa testuale che definisce il tempo in secondi di percorrenza del link
- **['linkpreview']**
Comando con cui viene richiesta una previsualizzazione di come il link (correntemente selezionato) verrà navigato.
- **['deletelink']**
Comando con cui viene richiesta la cancellazione del link correntemente selezionato.
- **['modify',path,text,font,dec_color]**
Comando con cui viene richiesta la modifica di un oggetto testuale.
Parametri:
<path> percorso dell'oggetto nel'IL
<text> contenuto testuale dell'oggetto
 tipo di carattere utilizzato
<dec_color> colore del testo espresso come vettore [r,g,b] con r,g,b espressi nell'intervallo da 0 a 255
- **['modify',<path>,<url>,<caption>]**
Comando con cui viene richiesta la modifica di un oggetto immagine.
Parametri:
<path> percorso dell'oggetto nel'IL
<url> il path del file nel server (es: /upload/images/imm.jpg)
<caption> stringa testuale che descrive il contenuto dell'immagine (didascalia)
- **['modify',<path>,<url>]**
Comando con cui viene richiesta la modifica di un oggetto ILMODEL.

Parametri:

<path> percorso dell'oggetto nel'IL

<url> il path del file nel server (es: /upload/3d_models/mod.aam)

- [**'mode', <boolean>**]
Comando con cui viene impostata la modalità di lavoro (editor o viewer) a seconda del valore del parametro <boolean>: se false la modalità è di editor, altrimenti è di viewer.

- [**'globalprop', <backgroundColor>, <startpos>**]
Comando con cui viene richiesta la modifica delle proprietà globali della scena come il colore di sfondo, o la posizione iniziale di camera.

Parametri:

<backgroundColor> colore della scena espresso come vettore [r,g,b] con r,g,b espressi nell'intervallo da 0 a 255

<startpos> è una stringa testuale nel formato "x,y,z" che esprime la posizione iniziale della camera

- [**'ilmapZoomIn'**]
[**'ilmapZoomOut'**]

Comandi utilizzati per richiedere un zoom-in o uno zoom-out nella ilmap.

- [**'ilmaplevel', <level>**]
Comando utilizzato per definire il livello gerarchico degli elementi che si vogliono visualizzare nella mappa.

Parametro:

<level> stringa testuale che indica quale livello gerarchico del paesaggio vogliono visualizzare nella mappa. Può essere "sites", "pages", "objects"

- [**'set_transform', <pos>, <scale>, <rot>**]
Comando inviato dalla pagina web a XVR per settare i valori di posizione, rotazione e scalatura di un oggetto.

Parametro:

<pos> array contenente i valori di posizione nel formato ["x","y","z"]

<scale> array contenente i valori di scalatura nel formato ["x","y"]

<rot> array contenente i valori di rotazione nel formato ["angle","x","y","z"]

Nel caso un oggetto non presenti una di queste proprietà, il parametro relativo andrà impostato a "null"

APPENDICE B (Discorso introduttivo all'esperimento)

Grazie della disponibilità a partecipare a questo esperimento. Esso ha lo scopo valutare la facilità di uso e l'efficacia di un sistema per la visualizzazione e interazione con un particolare tipo di ambienti virtuali chiamati Paesaggi di Informazione (Information Landscapes).

Si tratta di un sistema innovativo che propone informazioni come testi ed immagini distribuiti in uno spazio tridimensionale che può essere navigato liberamente o seguendo dei percorsi predefiniti determinati da collegamenti tra i vari elementi.

Questo sistema può essere fruito sia usando i normali personal computer Windows, sia secondo una modalità immersiva ottenuta usando un particolare strumento di visualizzazione chiamato cave. Il test verrà svolto in entrambe le modalità.

In entrambi i casi il test inizierà con una sessione di addestramento durante la quale verrai addestrato nell'utilizzo del dispositivo di interazione (per es: mouse o joystick) e durante la quale avrai il tuo primo contatto con il sistema.

Seguirà il test vero e proprio, che consisterà in varie fasi durante le quali ti verrà chiesto di svolgere alcuni compiti usando questo sistema. Maggiori informazioni sulle varie fasi ti saranno fornite durante il test. In caso di necessità, potrai chiedere l'aiuto del valutatore che provvederà a supportarti.

Alla fine ti verrà chiesto di compilare un questionario di valutazione del sistema.

La sessione complessiva avrà una durata complessiva di circa 45 minuti/1 ora.

Prima di iniziare, hai qualche domanda?

APPENDICE C (Modulo per il consenso informato)

Modulo per il consenso informato

Con questo modulo ti informiamo che stai per prendere parte come partecipante ad un esperimento per la valutazione dell'usabilità di un sistema informatico per la visualizzazione ed interazione con i "Paesaggi di Informazione". La sessione di valutazione sarà condotta all'interno del laboratorio di Robotica Percettiva (PERCRO) della Scuola Superiore Sant'Anna, Via Alamanni, Ghezzano San Giuliano Terme (Pisa).

Durante il test sarai chiamato a svolgere alcuni compiti con questo sistema, esprimere opinioni o rispondere a domande. Ti ricordiamo che stiamo valutando il sistema per renderlo il più possibile efficace e facile da usare. Non stiamo in alcun modo valutando le tue capacità personali: non esistono quindi errori che puoi commettere né risposte corrette o sbagliate. Precisiamo inoltre che non sono conosciuti rischi associati con la partecipazione a questo esperimento.

In qualità di partecipante, ti vengono riconosciuti certi diritti che sono elencati qui sotto:

1. Il diritto ad abbandonare l'esperimento in qualsiasi momento e per qualsiasi ragione
2. I dati raccolti durante l'esperimento non saranno in alcun modo associabili alla tua identità grazie ai nostri sforzi per assicurare l'anonimato.
3. Ti preghiamo di non discutere della presente sessione di valutazione con le altre persone che devono ancora partecipare all'esperimento.

Ti ringraziamo per il tempo e le energie spese nel partecipare in questa valutazione.

La tua firma indica che hai letto questo modulo nella sua interezza e che intendi partecipare come volontario all'esperimento.

Nome _____

Data, Firma _____

APPENDICE D (Questionario finale dei test)

Durante il test sono stati utilizzati due questionari: uno per la valutazione dell'esperienza desktop, l'altra per l'esperienza immersiva. Per esigenze di spazio, essendo i questionari molto simili riportiamo solo il primo. Il questionario relativo all'esperienza immersiva differisce dal primo solo per il fatto che:

- le domande sono riferite al joystick anziché al mouse
- è presente una domanda aggiuntiva relativa al cybersickness (Hai manifestato disagio durante la visione dell'applicazione?)

Questionario (valutazione esperienza desktop)

Informazioni Generali

- 1) Sesso: M F
- 2) Età: _____
- 3) Titolo di Studio: _____

- 4) Livello di esperienza di uso di personal computer, dove 1 indica "Scarso" e 5 indica "Ottimo":
1 2 3 4 5

- 5) Livello di esperienza nell'utilizzo di applicazioni 3D interattive (es: videogames, realtà virtuale ecc.), da 1 a 5, dove 1 indica "Scarso" e 5 indica "Ottimo":
1 2 3 4 5

Tecnica di interazione (apprendimento)

- 6) Hai incontrato difficoltà nell'apprendimento della tecnica di utilizzo del mouse per interagire con l'ambiente?

A) No, nessuna particolare difficoltà

B) Sì, alcune difficoltà

C) Sì, molte difficoltà

7) Se hai risposto affermativamente alla domanda precedente, indica il tipo di difficoltà incontrate

8) Quanto ritieni accettabile il tempo necessario all'apprendimento della tecnica di interazione con il mouse, in una scala da 1 a 5 (dove 1="del tutto inaccettabile", 5="pienamente accettabile")

1 2 3 4 5

User satisfaction

Opinione generale

9) Qual è stata la tua impressione generale riguardo il sistema all'inizio del test?

- A) molto negativa
- B) negativa
- C) abbastanza negativa
- D) di indifferenza (né negativa, né positiva)
- E) abbastanza positiva
- F) positiva
- G) molto positiva

10) Qual è la tua opinione generale riguardo il sistema dopo che lo hai utilizzato durante il test?

- A) molto negativa
- B) negativa
- C) abbastanza negativa
- D) di indifferenza (né negativa, né positiva)
- E) abbastanza positiva
- F) positiva
- G) molto positiva

11) Se la tua opinione è cambiata rispetto all'inizio, a cosa pensi sia dovuto il cambiamento?

12) Utilizzeresti di nuovo questo sistema?

- si
- no
- forse

13) Per quali scopi ritieni che questo sistema possa essere adatto?

- Ludico
- Didattico
- Nella ricerca
- Nei musei
- In applicazioni di uso comune (come internet ecc)
- Altro _____

14) Se nell'attuale implementazione non ritieni questo strumento efficace, ritieni che abbia comunque potenzialità interessanti qualora venga migliorato?

- si
- no
- forse

15) Quali aspetti ritieni debbano essere migliorati?

- a. Aspetti di visualizzazione
- b. modalità di navigazione e interazione
- c. supporto per l'orientamento nell'ambiente (o altri aiuti)
- d. Altro: _____

Opinioni sulla modalità di visualizzazione

16) Quali vantaggi pensi abbia apportato questo metodo di visualizzazione?

- Maggiore valenza espressiva
- Rappresentazione più intuitiva delle informazioni
- Relazioni più immediate tra gli elementi
- Accesso più veloce alle informazioni
- Maggiore apporto di informazioni

17) Quali svantaggi pensi abbia apportato questo metodo di visualizzazione?

- Minore valenza espressiva
- Rappresentazione meno intuitiva delle informazioni
- Relazioni meno immediate tra gli elementi
- Minore apporto di informazioni

18) Pensi che questa metodologia di divulgazione delle informazioni abbia effettivamente migliorato e facilitato l'accesso ai contenuti o ne abbia costituito un ostacolo?

A) Ha migliorato l'accesso

B) Ha migliorato alcuni aspetti

C) Ha costituito un ostacolo

19) Pensi che la visualizzazione in tre dimensioni abbia consentito un accesso più completo alle informazioni o che abbia comportato solo un dispendio di risorse?

A) accesso completo

B) accesso completo con dispendio di risorse

C) solo un dispendio di risorse

20) Hai trovato difficoltà nella lettura dei testi in 3 dimensioni?

A) nessuna difficoltà

B) alcune difficoltà

C) molte difficoltà

21) Ritieni che la soluzione di utilizzare un quadratino sopra l'oggetto per indicare la presenza di un collegamento (link) sia:

a. efficace

b. non molto efficace

c. per niente efficace

22) Riassumi con un voto da 1 a 5 la tua opinione riguardo la modalità di visualizzazione (1="scarso", 5="ottimo")

1 2 3 4 5

Opinioni sulla tecnica di interazione

23) Una volta imparata, quanto ritieni efficace la tecnica di interazione con il mouse? (1="scarso", 5="ottimo")

1 2 3 4 5

24) Scegli una o più espressioni che meglio identificano l'interazione con il mouse

- intuitivo
- efficace
- semplice
- troppe combinazioni da ricordare
- poco intuitivo
- dispendioso

25) Se la tua opinione sulla tecnica di interazione non è positiva, indica dei suggerimenti su come potrebbe essere migliorata

26) Hai preferito la navigazione libera o la navigazione tramite selezione della destinazione che vuoi raggiungere?

- sono entrambe efficaci e necessarie
- ho preferito la navigazione libera
- ho preferito la navigazione tramite selezione
- ritengo che le due modalità di navigazione siano utili ma non ben integrate tra loro

WAYFINDING

Queste domande si riferiscono esclusivamente al test di orientamento avvenuto all'interno del paesaggio di informazione sulle arti grafiche lucchesi.

1. Ritieni di aver avuto difficoltà durante l'esplorazione dell'ambiente?
 - Nessuna rilevante difficoltà
 - Qualche difficoltà
 - Molte difficoltà
2. Se sì, a cosa ritieni possano essere dovute?
 - a. Difficoltà nel pilotare il movimento tramite dispositivo (mouse, joystick ecc)
 - b. Mancanza di esperienza
 - c. Mancano facilitazioni per l'orientamento nello spazio
 - d. Le informazioni non sono organizzate in modo chiaro nell'ambiente
3. Quante volte ti è capitato di sentirti perso?
 - a. mai
 - b. qualche volta
 - c. molte volte
4. Ritieni che la presenza di una mappa spaziale avrebbe potuto facilitarti l'orientamento?
 - a. Non credo

- b. Forse
 - c. sicuramente
5. Eri già a conoscenza da altre fonti della tematica affrontata nell'IL?
- a. Non so niente sull'argomento
 - b. Ne so qualcosa
 - c. Sono uno esperto di questo settore
6. Ritieni di aver compreso (almeno in parte) la struttura spaziale delle informazioni?
- a. Per niente
 - b. In parte
 - c. Totalmente

Spazio per eventuali osservazioni aggiuntive:

Bibliografia

Bowman, D., Kruijff, E., LaViola, J., & Poupyrev, I. (2004). *3D User Interfaces: Theory and Practice*. Boston: Addison-Wesley.

Card, S. (1991). Cone Trees: animated 3D visualizations of hierarchical information. *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*.

Card, S., Robertson, G., & York, W. (1996). The WebBook and the Web Forager: an information workspace for the World-Wide Web. *Proceedings of the SIGCHI conference on Human factors in computing systems: common ground*, (p. 111-ff.). Vancouver, British Columbia, Canada.

Carrozzino, M., & Bergamasco, M. (2010). Beyond virtual museums: Experiencing immersive virtual reality in real museum. *Journal of Cultural Heritage* 11 , 452,458.

Carrozzino, M., Evangelista, C., Ruffaldi, E., Neri, V., & Bergamasco, M. (2009). Web Dissemination of Cultural Content Through Information Landscapes. *Proceedings of the Museums and the Web 2009*. Indianapolis, United States.

Carrozzino, M., Tecchia, F., Bacinelli, S., Cappelletti, C., & Bergamasco, M. (2005). Lowering the development time of multimodal interactive application: the real life experience of the XVR project. *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*. Valencia, Spain.

Chen, J., Pyla, P. S., & Bowman, D. A. (2004). Testbed Evaluation of Navigation and Text Display Techniques in an Information-Rich Virtual Environment. *Proc Virtual Reality* .

Ditterline, P., Engel, B., & Yeung, B. (2000). The effects of kinetic typography on readability.

Dodge, M., & Kitchin, R. (2001). *Atlas of CyberSpace*. Addison Wesley.

Goldfarb, D., Arends, M., Froschauer, J., & Merkl, W. (2011a). Combining Cultural Heritage Related Web Resources in 3D Information Landscapes. *Electronic Visualization and the Arts (EVA)* , 73-80.

Goldfarb, D., Arends, M., Froschauer, J., & Merkl, W. (2011b). Revisiting 3D Information landscapes for the display of the historical web content. *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*; (p. Paper-Nr. 46, 8 S). Lissabon, Portugal.

Grossman, T., Wigdor, D., & Balakrishnan, R. (2007). Exploring and Reducing the Effects of Orientation on Text Readability in Volumetric Displays. *CHI* .

Jankowski, J., Samp, K., Irzynska, I., Jozwowitz, M., & Decker, S. (2010). Integrating Text with Video and 3D graphics: The Effects of Text Drawing Styles on Text Readability. *ACM CHI* .

Kuhn, W., & Blumenthal, B. (1996). Spatialization: Spatial Metaphors for User Interfaces. *Proceedings CHI '96 Conference companion on Human factors in computing systems: common ground*, (p. 346-347).

Lanier, J. (1989). A portrait of the young visionary. *Whole Earth Review, Fall* , 108-19.

Larson, k., Van Dantzich, M., Czerwinski, M., & G, R. (2000). Text in 3D: some legibility results. *ACM CHI* , 145-146.

- Mazza, R. (2009). *Introduction to Information Visualization*. London: Springer-Verlag.
- Minakuchi, M., & Kidawara, Y. (2008). Kinetic typography for ambient displays. *ACM* , 54.
- Nelson, T. H. (1992). *Literary Machines 90.1*. (V. Scaravelli, Trad.) Padova: Franco Muzio Editore.
- Nelson, T. H. (2006). Lost in hyperspace. *New Scientist magazine* , 26.
- Nelson, T. H., Smith, R. A., & Mallicoat, M. (2007). Back to the future: hypertext the way it used to be. *ProceedingsHT '07 of the eighteenth conference on Hypertext and hypermedia*, (p. 227-228). New York, USA.
- Polys, N. F., Bowman, D. A., & North, C. (2004). Information-Rich Virtual Environment: Challenges and Outlook. *NASA Virtual Iron Bird Workshop* .
- Ridi, R. (1995). XANADU: l'ipertesto globale tra utopia e realtà. "*Università: quale biblioteca? Atti del seminario-dibattito*", (p. 153-161).
- Robertson, G., Mackinlay, J. D., & Card, S. (1991). The Perspective Wall: Detail And Context Smoothly Integrated. *Proceedings of CHI '91 Conference*, (p. 173-179). New Orleans, Louisiana.
- Ruddle, R. A., & Lessels, S. (2006). Three levels of metric for evaluating wayfinding. *Presence: Teleoperators and Virtual Environments* , 637-654.
- Ruffaldi, E., Evangelista, C., Neri, V., Carrozzino, M., & Bergamasco, M. (2008). Design of Information Landscapes for Cultural Heritage. *Proceedings of the 3rd ACM International Conference on Digital Interactive Media (DIMEA08)*. Athens, Greece.
- Shoemake, K. (1992). ARCBALL: A User Interface for Specifying Three-Dimensional Orientation Using A Mouse. *Proc. Graphics Interface '92*. Vancouver.
- Small, D. (1996). Navigating large bodies of text. *IBM Systems Journal* , 514-525.
- Small, D., Ishizaki, S., & Cooper, M. (1994). Typographic Space. *CHI94 Companion*. Boston, Massachusetts USA.
- Snowdon, D., Fahlén, L. E., & Stenius, M. (1996). WWW3D: A 3D Multi-User Web Browser. *WebNet, AACE* .
- Sparacino, F. (2001). (Some) computer vision based interfaces for interactive art and entertainment installations. *INTER_FACE Body Boudaries, n.2*. Paris, France.
- Sparacino, F., Davenport, G., & Pentland, A. (2000). City of News. *KOS* , 179-180.
- Spence, R. (2001). *Information Visualization*. Reading, MA: Addison-Wesley.
- Spence, R. (2007). *Information Visualization, Design for Interaction*. Harlow: Pearson Education, 2nd edition.
- Tufte, E. R. (2006). *Beautiful Evidence*. Cheshire, CT: Graphics Press.
- Tufte, E. R. (1990). *Envisioning Information*. Cheshire, CT: Graphics Press.
- Tufte, E. R. (1983). *The Visual Display of Quantitative Information*. Cheshire: Graphics Press.
- Tufte, E. R. (1997). *Visual Explanations*. Cheshire, CT: Graphics Press.

Ware, C. (2004). *Information Visualization: perception for design*. Morgan Kaufmann, 2nd edition.

Wielemaker, J., Hildebrand, M., Van Ossenbruggen, J., & Schreiber, G. (2008). Thesaurus-Based Search in Large Heterogeneous Collections. *Proceedings of the International Semantic Web Conference* (p. 695-708). Berlin - Heidelberg: Springer-Verlag.