

UNIVERSITÀ DEGLI STUDI DI TRENTO
Facoltà di Scienze Matematiche, Fisiche e Naturali



Corso di Laurea triennale in Informatica

Elaborato finale

SVILUPPO DI UNO STRUMENTO DIDATTICO USABILE NEL
CONTESTO DELLE LAVAGNE INTERATTIVE MULTIMEDIALI:
UN GENERATORE DI CRUCIVERBA

Relatore: Marco Ronchetti

Laureando: Marco Diener

Anno Accademico 2006 - 2007

SOMMARIO

INTRODUZIONE	3
1. LE LAVAGNE INTERATTIVE	6
1.1. Lo strumento	6
1.2. Il progetto	8
2. LA FORMAZIONE ELETTRONICA	15
2.1. L'e-learning	15
2.2. I Learning Management System	15
2.3. I learning object	16
3. L'UTILITÀ DIDATTICA DEI CRUCIVERBA	21
4. CONCETTI E TECNOLOGIE UTILIZZATI	23
4.1. L'interfaccia utente	23
4.1.1. <i>L'interfaccia grafica</i>	24
4.1.2. <i>Altri tipi di interfaccia</i>	26
4.2. Java	27
4.3. Flash	33
5. L'APPLICAZIONE	36
5.1. Gauge	37
5.2. Il cruciverba in Flash	40
5.3. Nuove funzionalità	45
5.4. Le ultime modifiche	48
5.5. La struttura del file XML	52
5.6. L'applicazione Java	60
5.7. La riprogettazione dell'applicazione Java	82
5.8. Le ultime ulteriori modifiche	114
CONCLUSIONE	121
BIBLIOGRAFIA	123

INTRODUZIONE

Il mondo moderno è caratterizzato da grandi sviluppi tecnologici che si susseguono senza sosta e con costanza nei paesi sviluppati, tanto che pressoché ogni realtà è da ciò influenzata nel bene e nel male. Scuola ed istruzione non ne sono esenti, tanto che l'immagine delle lezioni in aula con il solo ausilio di lavagna e libro di testo sta invecchiando ed a volte rappresenta già il passato.

Le nuove tecnologie a scuola hanno trasformato la relazione comunicativa tra studenti ed insegnanti, modificando gli stili di apprendimento, le strategie formative e le metodologie educative attuate. Inoltre la sempre maggiore diffusione della connessione ad Internet a banda larga ha portato alla nascita del fenomeno *e-learning*, ovvero la formazione elettronica, di cui è possibile usufruire da un qualsiasi calcolatore, il quale ha portato al decentramento dell'istruzione e l'annullamento delle distanze.

Tra le tecnologie emergenti in ambito educativo scolastico, la *lavagna interattiva multimediale* (LIM) introduce un potenziale di innovazione radicale per la didattica e la comunicazione formativa. Si tratta di uno strumento per la proiezione e la manipolazione di contenuti digitali che consente l'interazione "a mano libera" toccandone la superficie con un dito o con una speciale penna. Questa tipologia di utilizzo, intuitiva e naturale, permette di fondere insieme sulla superficie di questa lavagna le caratteristiche dell'interazione su computer e le attività che insegnanti e docenti svolgono abitualmente sulla lavagna tradizionale: scrivere, schematizzare, disegnare, cancellare, ma anche utilizzare programmi, interagire con ipertesti, mappe, animazioni e simulazioni, navigare su internet, comunicare a distanza con altri utenti collegati.

La Provincia Autonoma di Trento ha deciso di avviare un progetto di modernizzazione delle sue scuole, concretizzato nell'*Iniziativa Lavagne Interattive* ideata dal Dipartimento di Informatica e Telecomunicazioni dell'Università degli Studi di Trento, il quale prevede l'installazione e l'integrazione di numerose lavagne virtuali in tutte le scuole della provincia sull'esempio di altre esperienze simili avvenute in ambito europeo.

L'introduzione di una simile tecnologia, e soprattutto il fatto di saper sfruttare appieno tutte le sue potenzialità, comporta una serie di conseguenze che vanno dall'acquisto ed installazione delle lavagne, allo studio di nuovi percorsi didattici che ne integrino l'utilizzo, fino alla ricerca ed allo sviluppo di oggetti didattici digitali, strumenti ed applicazioni informatiche che consentano di avvalorare tali potenzialità. Tutte queste problematiche sono state affrontate e sono tutt'ora attuali nell'ambito dell'*Iniziativa Lavagne Interattive*. Tutti gli attori coinvolti in questo progetto fanno la sua parte per portare avanti tale iniziativa permettendole di avere il miglior impatto possibile con la realtà scolastica provinciale.

Questa tesi è stata composta in seguito all'implementazione di una delle applicazioni didattiche poco sopra citate sviluppate dall'Università degli Studi di Trento, ed è quindi incentrata sulla sua descrizione. Si tratta di un programma progettato in modo tale da consentire ad insegnanti e studenti di costruire e quindi risolvere dei cruciverba interattivi che possono essere utilizzati ad esempio per l'insegnamento di specifici termini legati ad un contesto didattico o vocaboli di una lingua straniera.

La tesi si divide in cinque parti principali: si apre con la prima che presenta il contesto in cui ci si sta muovendo (1), descrivendo lo strumento didattico in esame, ovvero le lavagne interattive multimediali (1.1), e quindi l'*Iniziativa Lavagne Interattive* nell'ambito della Provincia Autonoma di Trento (1.2).

In seguito, nella seconda parte, vengono presentate altre realtà didattiche introdotte grazie allo sviluppo tecnologico moderno (2), che hanno rappresentato tra l'altro concetti chiave nella mia personale esperienza di stage universitario: si tratta del fenomeno *e-learning* (2.1) e degli strumenti principali che consentono la sua realizzazione pratica, ovvero le piattaforme *Learning Management System* (2.2), sistemi informatici che rendono possibile la gestione di corsi e contenuti didattici online, ed i *Learning Object* (2.3), cioè oggetti didattici digitali.

Nella terza parte si discute dell'*utilità didattica* che può avere uno strumento come il cruciverba e quindi delle ragioni che hanno portato all'ideazione dell'applicazione generatrice di cruciverba descritta in seguito in questa tesi (3).

La quarta parte rappresenta una sorta di introduzione alla principale descrizione dell'applicazione stessa, in quanto elenca ed analizza le principali tecnologie delle quali ci si è serviti per la sua realizzazione (4). Il primo concetto di cui si parla è quello di *interfaccia utente* (4.1) ed in particolare ci si sofferma sull'interfaccia grafica (4.1.1), elemento di primaria importanza nel contesto dell'applicazione implementata, ma anche su altre tipologie di interfaccia (4.1.2) che possono in generale caratterizzare il software. In seguito si parla delle tecnologie *Java* (4.2), ovvero il linguaggio di programmazione utilizzato per l'implementazione del generatore di cruciverba in analisi, e *Flash* (4.3), quella utilizzata per i prodotti del programma stesso, ovvero i cruciverba interattivi.

La quinta ed ultima parte infine descrive in dettaglio ogni aspetto dell'applicazione sviluppata (5). Si spiegato nel testo, questa sezione segue cronologicamente l'effettiva lavorazione al software e di conseguenza inizia con l'analisi del prodotto del generatore di cruciverba: i cruciverba interattivi stessi. Viene presentato uno strumento esterno denominato *Gaupe* utilizzato nella loro progettazione (5.1), quindi si passa alla loro descrizione dettagliata (5.2) che prosegue nelle sezioni successive (5.3 e 5.4) dove si osservano tutte le modifiche e le revisioni effettuate fino al raggiungimento della

versione finale dei cruciverba. Segue quindi un'analisi approfondita dell'elemento più importante che li riguarda, ovvero un file *XML* che sta alla base della loro realizzazione (5.5). A questo punto l'attenzione si sposta sull'applicazione che li dovrà generare (5.6) ed anch'essa viene descritta fin nei minimi dettagli. Di tale programma sono state implementate due versioni, la seconda delle quali risulta essere un'intera riprogettazione della precedente, migliorata sotto ogni punto di vista ed alla quale sono state aggiunte numerose funzionalità, come viene spiegato nella sezione successiva (5.7). Anche quest'ultima versione ha però subito revisioni e modifiche al termine del suo sviluppo, ed esse sono tutte descritte nell'ultima parte del testo (5.8), al termine della quale corrisponde la fine dei lavori sull'applicazione.

1. LE LAVAGNE INTERATTIVE

1.1. Lo strumento

Una *lavagna interattiva* (detta anche LIM, acronimo di Lavagna Interattiva Multimediale, e IWB in inglese, acronimo di Interactive WhiteBoard) è uno schermo di proiezione che al tempo stesso è un dispositivo sensibile al tocco. Collegata ad un computer e tramite una specifica tecnica di visualizzazione (come ad esempio l'uso di un proiettore), opera come un grande schermo con cui insegnanti e studenti interagiscono mediante il semplice tocco delle dita o di una speciale penna come se usassero il mouse del computer. Con questa modalità, dallo schermo della lavagna è possibile controllare ed accedere ad ogni applicazione, file e contenuto multimediale, navigare su Internet, far girare CD-ROM e DVD e quant'altro, estendendo in tal modo il classico paradigma didattico centrato sulla lavagna.

Le LIM vengono collegate al computer in cui viene caricato il software necessario per la comunicazione tra lavagna e calcolatore tramite cavo (porta seriale o USB) oppure tramite una connessione wireless (Bluetooth). Vi sono inoltre tre principali tecnologie differenti per quanto riguarda la gestione dell'interazione al tocco sulla superficie della lavagna:

- **Analogico-resistiva:** quando si tocca la superficie della lavagna si provoca in essa una lieve deformazione che spinge due strati di materiale conduttore elettrico a toccarsi, generando un segnale elettrico che determina le coordinate alle quali la pressione è stata applicata;
- **Elettromagnetica:** una penna speciale produce un segnale elettrico, poi tradotto in coordinate, quando essa viene a contatto con la griglia digitalizzata di cui è dotata la lavagna. L'uso di tale penna è necessario, il semplice tocco delle dita non ha effetto. Vi sono due varianti di questa tecnologia:
 - *Attiva:* la penna è alimentata da una fonte energetica e produce attivamente un segnale elettrico, aumentando la velocità di risposta al tocco;
 - *Passiva:* la penna non è alimentata e si limita a chiudere il circuito elettrico a contatto con la lavagna elettrizzata.
- **A triangolazione:** le coordinate vengono determinate triangolando i dati ottenuti da sensori posti ai lati o agli angoli della lavagna. Vi sono tre mezzi utilizzati per fornire dati ai sensori (in ogni caso è necessario l'uso di una speciale penna):
 - *Laser:* raggi laser a bassa potenza spazzano la superficie della lavagna. La penna è dotata di superficie riflettente in grado di riflettere i raggi alle sorgenti;

- *Ultrasuoni*: quando viene premuta sulla superficie della lavagna, la penna emette degli ultrasuoni rilevati da appositi microfoni;
- *Infrarossi*: quando viene premuta sulla superficie della lavagna, la penna produce luce infrarossa rilevata da appositi sensori ottici.

Il vantaggio immediato della tecnologia resistiva è la possibilità di usare il tocco delle dita senza necessità di una penna speciale, mentre quelli delle altre due tecnologie sono la possibilità di usare la lavagna anche in modo tradizionale e la robustezza di quest'ultima.

Le LIM si differenziano anche per la tecnica di visualizzazione dell'immagine; ve ne sono di due tipologie:

- **A proiezione frontale**: queste lavagne necessitano del supporto di un proiettore di fronte ad esse. Presentano il disagio procurato dal fascio di luce prodotto dal proiettore negli occhi dell'insegnante rivolto agli ascoltatori e quindi al proiettore stesso, inoltre vedono proiettata su di esse l'ombra di un insegnante che staziona tra esse ed il proiettore. Per ridurre tali svantaggi alcuni produttori incorporano alle LIM dei sistemi di proiezione ravvicinata e ad angoli di circa 45°;
- **A retroproiezione**: queste lavagne possiedono un proiettore incorporato dietro ad esse, prevenendo gli svantaggi della proiezione frontale. Questi sistemi sono però più ingombranti presentando uno spessore maggiore, non è possibile appenderli alle pareti e sono tipicamente più costosi.

Il software incluso nell'acquisto delle lavagne interattive fornisce usualmente il supporto di alcuni strumenti per sfruttare al meglio tale tecnologia e per stimolare l'interazione; quello principale consente di scrivere con dell'inchiostro digitale su uno strato virtualmente sovrapposto alla normale immagine mostrata dal computer. Note, appunti e schemi così realizzati possono inoltre essere salvati e condivisi.

Secondo gli studi, come ad esempio le ricerche svolte dalla britannica BECTA (British Educational Communications and Technology Agency)¹, rispetto ad altre tecnologie normalmente usate in classe, come ad esempio l'uso del solo computer, la lavagna interattiva consente di aumentare l'attenzione, il coinvolgimento e la motivazione degli studenti, e di migliorarne l'apprendimento in

¹ Vedere "What research says about interactive whiteboards" pubblicato dalla BECTA nel 2003 e disponibile al link http://www.becta.org.uk/page_documents/research/wtrs_whiteboards.pdf

termini di comprensione di concetti più complessi grazie alle presentazioni più chiare, efficienti e dinamiche, di capacità di rapportarsi a diversi stili di apprendimento stesso e di creatività nell'ideare presentazioni. Avvantaggia inoltre gli studenti più giovani e quelli disabili grazie all'indipendenza dall'uso di strumenti quali la tastiera.

1.2. Il progetto

In ambito locale, l'Assessorato all'istruzione e alle politiche giovanili della Provincia Autonoma di Trento ha deciso l'acquisto di un numero elevato di LIM da installare nelle scuole di ogni ordine e grado della Provincia.

Si tratta in particolare di lavagne prodotte dalla SMART Technologies, modelli della serie *SMART Board 600* a proiezione frontale, dotate di tecnologia analogico-resistiva per quando riguarda la digitazione. Si connettono ad un computer tramite porta USB 2.0, dalla quale vengono anche alimentate². Queste lavagne sono dotate di quattro penne speciali colorate (nero, blu, verde e rosso) e di un cancellino rettangolare; un sensore ottico rileva quando uno di questi strumenti viene sollevato dal suo supporto, comunicando alla lavagna il relativo colore da utilizzare nelle successive scritture sulla superficie della lavagna, oppure, nel caso del cancellino, la richiesta di cancellare le note già presenti sullo schermo. È presente anche un pulsante che emula la pressione del tasto destro del mouse.

Il software fornito insieme a queste LIM include numerosi utili strumenti per rendere l'uso della lavagna e le presentazioni le più avanzate possibili; ecco i principali:

- **Modalità di interazione:** è possibile scegliere la modalità di interazione con la lavagna tramite tocco; vi è la modalità cursore (con la quale si emula l'uso del mouse) e quella di scrittura. È inoltre possibile realizzare diversi tipi di scrittura, tra cui i principali sono quello normale e quello semitrasparente (modalità evidenziatore);
- **Notebook:** con questa applicazione è possibile salvare le note scritte sulla lavagna ed organizzarle in pagine, creando così una presentazione. È inoltre possibile importare oggetti (come immagini, video, contenuti in formato Flash) memorizzati in una particolare "galleria" ed esportare le presentazioni create in formati HTML, PDF o PowerPoint;
- **Registratore:** con esso è possibile registrare un video con tutte le azioni svolte sulla lavagna, con la possibilità di integrarvi anche l'audio tramite un microfono;

² Per maggiori informazioni visitare <http://www2.smarttech.com/st/en-US/Support/FrontProjection/600series.htm>

- **Riproduttore video:** con questo strumento è possibile visualizzare dei video, scrivere e disegnare su di essi e catturarne dei singoli fotogrammi;
- **Tastiera su schermo:** viene visualizzata una tastiera sensibile al tocco sullo schermo, così da rendere possibile la digitazione senza allontanarsi dalla lavagna;
- **Strumenti fluttuanti:** si tratta di un'utilità che consente l'accesso rapido agli strumenti ed ai comandi di uso più frequente (è possibile scegliere quali inserire) tramite dei pulsanti sullo schermo;
- **Conferenza online:** dà la possibilità di partecipare o di creare una conferenza online;
- **Ombreggiatura schermo:** si possono creare aree ombreggiate sullo schermo con la possibilità di spostarle trascinandole; è possibile ad esempio spostare tale ombra rivelando gradualmente i contenuti sottostanti;
- **Luce mirata:** questo strumento dà la possibilità di ombreggiare l'intero schermo con l'eccezione di un'area che si vuol mettere in risalto (per focalizzarvi l'attenzione);
- **Lente di ingrandimento:** dà la possibilità di selezionare un'area dello schermo e di visualizzarla ingrandita in un'apposita finestra;
- **Puntatore:** con questo strumento è possibile posizionare delle frecce sullo schermo per dirigere l'attenzione verso informazioni interessanti;
- **Cattura dello schermo:** è possibile catturare un'istantanea di un'area dello schermo (o dell'intero schermo) che verrà automaticamente salvata in una pagina dell'applicazione Notebook;
- **Supporto per applicazioni Microsoft Office:** se tale suite è installata sul computer, verranno integrate in essa alcuni utili comandi. In particolare nelle applicazioni Word ed Excel è possibile inserire nel documento su cui si sta lavorando delle note scritte sulla lavagna in forma di immagine o di testo (il software è in grado di capire la scrittura manuale e tradurla in testo digitale) e acquisire istantanee della schermata (che vengono salvate in una pagina del software Notebook); nell'applicazione PowerPoint, durante la visualizzazione di una presentazione, vengono mostrati due pulsanti per navigare attraverso le diapositive ed uno per accedere ad un menù contenente ulteriori comandi, tra cui l'inserimento di note nel documento come immagini e la loro cancellazione, l'acquisizione di istantanee dello schermo, la creazione di una nuova diapositiva, la visualizzazione del menù di PowerPoint e la chiusura della presentazione.

Il Dipartimento di Informatica e Telecomunicazioni dell'Università di Trento ha proposto un progetto pluriennale di accompagnamento dell'introduzione delle lavagne denominato *Iniziativa Lavagne Interattive*, volto a:

- costruire una comunità attorno alle idee di insegnamento basato sulle lavagne interattive;
- effettuare ricerche metodologiche sull'uso delle Lavagne Interattive nel contesto didattico;
- effettuare ricerche tecnologiche legate all'uso delle Lavagne Interattive.

In generale l'iniziativa vuole raccogliere tutti gli attori che a vario titolo ruotano attorno all'uso delle lavagne interattive.

Un'istanza di tale iniziativa è rappresentata dal progetto trimestrale denominato *SLIM4DIDA* (acronimo di Supporto all'introduzione di Lavagne Interattive Multimediali per (4) la DIDAttica) finanziato dalla Provincia Autonoma di Trento (PAT), svoltosi dal febbraio all'aprile 2007. Gli attori da esso coinvolti sono:

- Informatica Trentina;
- Servizio per lo Sviluppo e l'Innovazione del Sistema Scolastico della PAT;
- Università di Trento (Dipartimento di Informatica e Telecomunicazioni);
- Computer Learning (attività di addestramento all'uso della LIM);
- Edulife (studio pedagogico relativo alle opportunità di utilizzo delle LIM).

Hanno partecipato al progetto i seguenti istituti scolastici individuati dal Servizio per lo Sviluppo e l'Innovazione del Sistema Scolastico della PAT:

- Istituto Bertrand Russell di Cles;
- Istituto Martino Martini di Mezzolombardo;
- I.T.C.G. Carlo Antonio Pilati di Cles;
- Liceo Scientifico e Linguistico Leonardo da Vinci di Trento;
- Liceo Scientifico Galileo Galilei di Trento;
- Istituto Comprensivo Trento 5;
- Istituto Comprensivo Bernardo Clesio di Cles.

Le attività previste nell'ambito del progetto SLIM4DIDA sono state:

- Incontri formali per la pianificazione del progetto e per la discussione dei progressi dello stesso;
- La formazione all'uso delle LIM per i docenti delle scuole che hanno aderito al progetto;
- L'esame delle modalità e di uno scenario d'uso di tale tecnologia;
- L'ideazione e la creazione di oggetti didattici (in inglese LO, Learning Object) in grado di sfruttare le potenzialità delle LIM e l'analisi dell'adeguatezza e dell'adattabilità del materiale didattico preesistente;
- La creazione di una comunità per la condivisione delle esperienze e della conoscenza, in particolare tramite il supporto di strumenti telematici;
- La valutazione di programmi didattici che includono l'uso di LO e LIM.

Durante gli incontri per la pianificazione del progetto sono state proposte delle attività didattiche che sfrutterebbero l'interattività e multimedialità delle lavagne virtuali; alcuni esempi comprendono:

- L'utilizzo di strumenti per la composizione di scritture collaborative;
- Il supporto di software come Google Earth nello studio della geografia e dell'architettura di specifici edifici;
- L'arricchimento dei contenuti didattici preesistenti attingendo immagini e contenuti audio e video da sorgenti multimediali online;
- L'utilizzo di software come Cabri e Maple per integrare lo studio della matematica;
- L'utilizzo di strumenti come i traduttori online nello studio delle lingue straniere (e l'analisi critica sulla loro qualità);
- L'uso di strumenti contenenti giochi, quiz e parole crociate;
- La sperimentazione di lezioni svolte tra classi remote tramite l'utilizzo della funzionalità videoconferenza fornita dalle LIM.

Il DIT (Dipartimento di Informatica e Telecomunicazioni) dell'Università di Trento si è occupato dell'analisi e dello sviluppo di *Learning Object* e di altri strumenti volti a rendere attuabili alcune delle attività appena citate e di altre ritenuti utili nello stesso ambito, compilando una scheda descrittiva per ognuno di essi. Alcuni esempi sono i seguenti:

- **Learning Object per la costruzione di linee temporali:** con esso è possibile visualizzare una linea del tempo ed inserirvi fatti ed eventi, così da dare un immediato quadro generale di un contesto temporale storico o moderno; vi sono già svariati strumenti online liberamente utilizzabili per svolgere questo compito, più o meno complessi e personalizzabili³;
- **Learning Object per la registrazione e la condivisione di formule matematiche:** con esso è possibile digitare e salvare delle formule matematiche per poi renderle sempre disponibili alla visualizzazione; online vi è uno strumento liberamente accessibile che svolge questo compito⁴. Vi sono già state inserite numerose formule, inoltre è anche possibile utilizzarle per ottenere risultati numerici;
- **Learning Object per la visualizzazione della struttura di edifici interessanti:** è stata presa come esempio la tomba del faraone Tutankhamon e, come per altre opere architettoniche importanti, è stato trovato online un plug-in per il software gratuito Google Earth in grado di consentirne la visita completa⁵;
- **Learning Object per la tracciatura di funzioni matematiche:** dà la possibilità di tracciare graficamente l'andamento di una o più funzioni matematiche lungo gli assi cartesiani. Esistono strumenti online in grado di fare ciò, inoltre ve ne sono anche numerosi liberamente scaricabili⁶;
- **Learning Object per l'analisi di formule e funzioni matematiche note:** con esso è possibile avere un immediato riscontro visivo del comportamento delle funzioni matematiche al variare dei parametri dai quali sono descritte. Anche in questo caso esistono degli strumenti online liberamente accessibili creati per svolgere tale compito⁷;
- **Learning Object per associare parole ed immagini:** tale LO fornisce la possibilità di associare parole diverse a parti di un'immagine, trascinandole nella zona corretta, utile quindi ad esempio per studiare la struttura di vari oggetti. Vi sono strumenti online per svolgere tale funzione, ma tipicamente poco personalizzabile, perciò ne è stato sviluppato uno appositamente per facilitarne il settaggio da parte degli insegnanti;
- **Learning Object per generare e risolvere cruciverba:** esso permette agli insegnanti di costruire facilmente dei cruciverba per farli poi risolvere agli studenti in modo da far apprendere loro in modo divertente termini specifici o in lingua straniera. Vi sono molte

³ Vedere ad esempio quelle alle pagine <http://www.free-timeline.com/timeline.jsp> e <http://timeline.cer.jhu.edu/index.htm>

⁴ Vedi pagina <http://instacalc.com/>

⁵ Vedere il link <http://google-mapping.blogspot.com/2007/02/la-tomba-di-tutankhamon-3d-in-google.html>

⁶ Vedere ad esempio quello alla pagina http://www.walterzorn.com/grapher/grapher_e.htm

⁷ Vedere ad esempio quelli sul sito della società giapponese IES Inc. al link <http://www.ies.co.jp/math/indexeng.html>

applicazioni online simili, ma solitamente si concentrano su una sola delle due funzionalità (creazione o risoluzione dei cruciverba) oppure non consentono una completa personalizzazione del cruciverba stesso, o non presentano ottime modalità di controllo; si è scelto quindi di sviluppare un'applicazione che risolva tutte le problematiche appena elencate. Questa è in particolare l'applicazione da me personalmente sviluppata su progetto del professor Marco Ronchetti del DIT dell'Università di Trento e rappresenta la natura della mia collaborazione nell'ambito dell'Iniziativa Lavagne Interattive.

Alcuni strumenti online utili per il progetto SLIM4DIDA sono rappresentati dal sistema Wiki gestito dal professor Marco Ronchetti⁸, coordinatore dell'Iniziativa Lavagne Interattive, e dalla piattaforma Didapat, un sistema LMS online che ospita anche una sezione riservata al progetto SLIM4DIDA. Didapat è stata sviluppata dalla collaborazione dei seguenti enti:

- Edulife Spa;
- Centro Pedagogico per l'Orientamento e la Formazione (Cepof);
- Istituto Salesiano Don Bosco ed Istituto Salesiano San Zeno;
- Istituto Superiore Internazionale Salesiano per la Ricerca Educativa (ISRE);
- Scuola di Specializzazione per l'Insegnamento Secondario Veneto (SSIS Veneto);
- Sinform Srl.

Il suo scopo è quello di favorire:

- lo sviluppo delle competenze degli insegnanti e dei formatori nell'utilizzo delle ICT (Information and Communications Technology) a scopo didattico e gestionale;
- il miglioramento dell'offerta formativa con l'adozione di nuovi approcci a livello pedagogico, organizzativo e territoriale;
- L'adeguamento delle dotazioni tecnologiche e potenziamento delle opportunità di cooperazione tra istituti e tra insegnanti a livello territoriale.

Per l'ambito SLIM4DIDA la finalità è stata quella di rendere disponibile un ambiente per capitalizzare le esperienze ed i documenti relativi alle LIM. Inoltre essa, grazie al servizio Forum, mette a disposizione un ambiente di condivisione delle analisi e delle riflessioni relative a questa esperienza.

⁸ Per maggiori informazioni visitare il link http://laurin.science.unitn.it/wiki_lavagne/index.php/Pagina_principale

Durante lo svolgimento del progetto si sono affrontate delle inevitabili difficoltà a partire dall'aspetto tecnico (in particolare durante il montaggio e l'utilizzo dei sistemi composti da computer, lavagna interattiva e proiettore) fino a quello concettuale (a causa delle difficoltà degli insegnanti nell'uso dei nuovi strumenti, ad esempio nello sfruttarli per incrementare l'interazione degli studenti così da stimolarli maggiormente, e nell'ideazione di oggetti didattici in grado di favorire tali potenzialità).

Tali problematiche erano state previste in anticipo grazie ad esperienze precedenti, come ad esempio quella rappresentata dall'iniziativa inglese denominata *SWE* (Schools interactive Whiteboard Expansion), svoltasi nell'anno 2003/04 in collaborazione con la BECTA, che rientra nel più ampio ambito del London Challenge, un accordo quinquennale (2001/06) tra il governo inglese, le scuole e la città di Londra con l'obiettivo di innalzare gli standard dell'istruzione cittadina. Il progetto SWE consisteva nell'installazione di lavagne interattive in tutte le scuole secondarie londinesi e nell'incorporare il loro utilizzo nello studio di matematica, scienze ed inglese.

È stata compilata un'analisi statistica, pubblicata nel gennaio 2007 dal DfES (acronimo di Department for Education and Skill) del governo britannico⁹, con lo scopo di valutare a posteriori il successo di tale progetto in termini di miglioramento nella qualità complessiva dell'istruzione nelle tre materie interessate, prendendo in considerazione un'ampia gamma di fattori. Tale studio ha permesso di ottenere dei risultati derivanti dal confronto dei dati relativi alle scuole londinesi con quelli relativi a scuole di altre aree non interessate dal progetto, ed essi mettono in luce dei miglioramenti, laddove presenti, di rilevanza assolutamente non significativa. L'SWE ha quindi rappresentato un fallimento? C'è da dire che tale studio non pretende di essere perfettamente preciso, poiché alcuni dati specifici non sono stati di facile reperimento (permettendo la presenza di errori) ed inoltre il campione di scuole preso in esame non è stato particolarmente ampio.

Indubbiamente le difficoltà che si incontrano lungo il cammino sono molte, ma le potenzialità di interazione e coinvolgimento delle lavagne interattive rispetto ai tradizionali metodi sono tuttavia innegabili, tanto che il progetto SWE ha avuto dei successori. Al giorno d'oggi le lavagne interattive sono una dotazione standard nelle scuole primarie e secondarie del Regno Unito.

⁹ Si tratta di "The Interactive Whiteboards, Pedagogy and Pupil Performance Evaluation: An Evaluation of the Schools Whiteboard Expansion (SWE) Project: London Challenge" disponibile al link <http://www.dfes.gov.uk/research/programmeofresearch/projectinformation.cfm?projectid=14213&resultspage=1>

2. LA FORMAZIONE ELETTRONICA

2.1. L'e-learning

Lo sviluppo delle nuove tecnologie dell'informazione e della comunicazione ha avuto significative ripercussioni anche sulle modalità di apprendimento, stimolando la formazione di nuove risorse didattiche. Ciò ha portato alla nascita del concetto di *e-learning*, ovvero l'apprendimento tramite l'impiego delle tecnologie informatiche e telematiche, dove molti dei materiali in formato cartaceo sono stati sostituiti da quelli di natura digitale.

C'è da dire che l'e-learning, oltre alla sua modalità "pura", ovvero svolta interamente tramite supporti digitali spesso online, può inserirsi in processi formativi definiti "misti" (si parla di *blended learning*) in cui la componente prettamente digitalizzata si affianca alla formazione di stampo tradizionale (interventi in aula, supporto telefonico, workshop, seminari, ecc.).

Due dei principali elementi che stanno alla base dell'e-learning sono ovviamente i contenuti ed inoltre, estremamente importanti, le comunità di pratica che si vengono a creare, ad esempio tramite un forum online, e che si accrescono con le esperienze di tutti i partecipanti.

Gli strumenti comunicativi sfruttati dall'e-learning sono di due tipi:

- **Sincroni**: richiedono la presenza al calcolatore in un particolare momento per prendere parte alla sessione didattica, come chat e videoconferenze;
- **Asincroni**: non richiedono alcun tipo di intervento in momenti precisati, come e-mail, forum e newsletter.

2.2. I Learning Management System

Quando si parla di e-learning si pensa normalmente anche ai *sistemi LMS* (acronimo di Learning Management System) che sono lo strumento tecnologico più usato per realizzarla nella pratica.

Si tratta di una tipologia di sistemi software studiati per consentire la gestione di contenuti didattici da parte degli insegnanti e per renderli disponibili agli studenti in modalità e-learning. In pratica presidiano la gestione e la distribuzione di corsi formativi online.

L'LMS dovrebbe implementare e rendere facilmente accessibili elementi come i seguenti per ogni corso: il programma di studio, dati amministrativi come il valore in crediti del corso ed i prerequisiti, uno spazio dove tenere aggiornate le informazioni varie riguardanti il corso, la registrazione degli utenti a tale corso (eventualmente gestendo un sistema di pagamento per l'iscrizione), il tracciamento delle attività on-line e dei progressi degli studenti, il materiale

didattico, ulteriori risorse come letture e link a materiale online, test di autovalutazione per gli studenti, supporti per la comunicazione elettronica come forum, chat ed e-mail, diritti di accesso diversi per insegnanti e studenti (solo gli insegnanti possono ad esempio caricare e modificare il materiale didattico), documentazione e statistiche relative al corso stesso. Tali elementi dovrebbero essere forniti dall'LMS tramite un'interfaccia grafica intuitiva, la quale deve inoltre permettere in maniera semplice il passaggio tra i diversi corsi gestiti dallo stesso sistema.

2.3. I learning object

Un *Learning Object* (LO) è definito dalla IEEE (Institute of Electrical and Electronics Engineers) come una qualsiasi entità, digitale o non, che possa essere usata, ri-usata o referenziata durante l'apprendimento/formazione supportato dalla tecnologia¹⁰. Questa definizione, ritenuta da alcuni troppo generica, è stata riformulata da David A. Wiley: un LO è una qualsiasi risorsa digitale che può essere riutilizzata per supportare l'apprendimento¹¹.

Un Learning Object è un contenuto didattico, un'unità di istruzione per l'e-learning. Le caratteristiche che devono essere prese in considerazione nella loro progettazione sono le seguenti:

- **L'autoconsistenza:** il LO deve risultare esaustivo per l'utente riguardo all'obiettivo formativo per cui è stato ideato;
- **La modularità:** il LO deve prestarsi ad essere associabile ad altri che rientrano nello stesso contesto didattico, rappresentando quindi un singolo modulo in cui tale contesto viene suddiviso. In questo modo l'utente può dedicare lassi di tempo anche brevi alla sua formazione nel contesto didattico interessato, personalizzando così tempi e modalità di approccio ai contenuti;
- **La reperibilità:** il LO deve essere descritto da dei metadati in esso contenuti (spesso organizzati in un foglio XML), in grado di classificarlo dettagliatamente. L'utilità di ciò non è solo quella di ottenere dati sul LO, ma anche quella di essere in grado di svolgere ricerche mirate per particolari LO negli appositi repository, ovvero archivi digitali online dove gli LO vengono raccolti e messi a disposizione. Esistono degli standard che pongono le linee

¹⁰ Vedere “IEEE Standard for Learning Object Metadata. IEEE Standard 1484.12.1” (2002), pubblicato da IEEE

¹¹ Vedere “Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy” (2000), editato da David A. Wiley, la cui versione online è disponibile al link <http://www.reusability.org/read/>

guida per la redazione dei metadati in modo da renderli accurati e completi; uno dei più autorevoli è quello chiamato LOM (Learning Object Metadata);

- **La riusabilità:** il LO deve essere versatile in modo da poterlo utilizzare in diverse situazioni di apprendimento ed in diversi contesti didattici;
- **L'interoperabilità:** il LO deve essere predisposto per poter essere distribuito su qualsiasi piattaforma LMS. A tal fine sono stati individuati degli standard che devono essere rispettati durante lo sviluppo del LO per garantire la comunicazione tra esso e qualsiasi piattaforma LMS in grado di supportare lo standard scelto. Attualmente lo standard che va per la maggiore è SCORM (acronimo di Sharable Content Object Reference Model).

Per garantire un'adeguata modularità del contesto didattico e la riusabilità del LO, seppur mantenendone l'autoconsistenza, è importante scegliere delle *dimensioni* ottimali per i contenuti di un singolo LO. La CISCO Systems ha dato un'indicazione in merito¹², stabilendo una possibile misura di grandezza ottimale degli LO stimata intorno ai 7 (+/-2) concetti da presentare al fruitore.

Quanto ai *metadati*, come già detto essi vengono compilati seguendo uno schema concettuale descritto da degli appositi standard come ad esempio il già citato LOM, il MARC (acronimo di MACHine-Readable Cataloging) ed il Dublin Core.

Analizzando ad esempio la struttura presentata dal LOM, sono state individuate nove aree descrittive contenenti gruppi di attributi strutturati ad albero¹³. Le aree sono le seguenti:

- **Generale:** conta 8 campi principali, come ad esempio “titolo” e “descrizione”, tramite i quali presenta le informazioni generali sul LO;
- **Ciclo di vita:** conta 3 campi principali e fornisce informazioni sulla storia del LO, sul suo stato attuale e sulle entità che vi hanno operato;
- **Meta-metadati:** conta 4 campi principali e descrive i metadati stessi, con informazioni ad esempio sullo schema adottato per redigerli e sugli autori;
- **Aspetto tecnico:** conta 7 campi principali e fornisce informazioni tecniche sul LO, come ad esempio il formato dei componenti in esso contenuti, le sue dimensioni totali, i requisiti tecnici da esso richiesti;

¹² Vedere l'articolo “Reusable Learning Object Strategy: Designing and Developing Learning Objects for Multiple Learning Approaches” pubblicato nel 2003 da CISCO Systems Inc. disponibile al link http://www.e-novalia.com/materiales/RLOW_07_03.pdf

¹³ Per maggiori informazioni vedere “Draft Standard for Learning Object Metadata”, pubblicato nel 2002 da IEEE e disponibile al link http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

- **Aspetto didattico:** conta 11 campi e descrive la natura didattica del LO, contenendo dati come il tipo di interattività fornita, il livello di difficoltà dei contenuti ed il contesto dei contenuti stessi;
- **Diritti:** conta 3 campi e fornisce informazioni sui diritti intellettuali e sulle condizioni d'uso per il LO;
- **Relazioni:** conta 2 campi principali tramite i quali si elencano le relazioni del LO con altre risorse;
- **Annotazioni:** conta 3 campi e contiene commenti sull'utilizzo del LO in ambito didattico, con informazioni sugli autori di tali commenti;
- **Classificazione:** conta 4 campi principali in cui si inseriscono le informazioni necessarie per la classificazione del LO all'interno di un particolare sistema di classificazione.

Lo standard *SCORM*, ufficializzato nel 2000 dall'ADL Initiative (Advanced Distributed Learning Initiative), specifica invece gli standard per l'interoperabilità dei LO.

Esso definisce l'insieme delle procedure che aggregano i contenuti dei LO ed il modo di elaborare questi contenuti sulle piattaforme LMS sulle quali i LO sono presentati: i dati riguardanti i LO e l'interazione tra essi ed i discenti sono gestiti da un file xml che garantisce l'interoperabilità. Le principali caratteristiche che vengono prese in considerazione dallo SCORM sono le seguenti:

- **La catalogabilità:** ottenuta tramite la specifica dei metadati, in accordo con lo standard LOM, in modo da permettere al LO di essere indicizzato e ricercato all'interno della piattaforma in cui si trova. I metadati devono essere archiviati in un file chiamato *imsmanifest.xml*;
- **Il dialogo con le piattaforme LMS:** il LO deve essere in grado di comunicare all'LMS dei dati utili al tracciamento dell'attività del discente, ad esempio il tempo passato allo studio di una certa lezione ed i vincoli che interessano il passaggio all'oggetto successivo. Normalmente il linguaggio usato per tale comunicazione è il javascript;
- **La riusabilità:** il LO deve essere trasportabile su qualsiasi piattaforma compatibile con SCORM senza perdita di funzionalità. Questo principio è alla base dello standard in quanto, rispettando le direttive di costruzione, il LO e la piattaforma non devono essere modificati per attivare le funzionalità di tracciamento e catalogazione.

Un oggetto SCORM è un pacchetto, un file con estensione .zip oppure .pif, che contiene al suo interno il LO vero e proprio ed i necessari metadati. SCORM non specifica un particolare formato

di file che debba rappresentare il LO; qualsiasi formato può essere incluso nel pacchetto, a patto che sia possibile applicarvi la tracciabilità (su alcuni formati ciò non è possibile). Tra i formati più comuni si può citare HTML, Flash e Java.

Gli elementi che compongono il pacchetto SCORM sono detti *resources*, ed ognuno di essi può essere a sua volta composto da più oggetti, come ad esempio una pagina HTML che contiene a sua volta delle immagini.

SCORM prevede la creazione di oggetti che, come detto, sono deputati a comunicare con l'LMS, i quali sono detti *SCO*, e di altri di supporto che non hanno questo requisito, detti *ASSET*. I dati che gli SCO possono comunicare all'LMS sono in pratica degli stati (per esempio “navigato”, “visualizzato ma non completato”, “finito”, “passato”, “non passato”, ecc.) e delle informazioni (come il tempo impiegato nella sessione, il nome dello studente, ecc.).

Gli oggetti SCORM possono anche presentare delle *organizations*, ovvero dei vincoli tra di essi. Per esempio può non essere consentito l'accesso ad un oggetto SCORM se prima non se ne completa un altro.

Analizzando i LO da un'ottica pedagogica orientata verso un approccio costruttivista all'apprendimento, vale innanzitutto la pena esplicitare i concetti chiave sui quali è basata la filosofia alla base dei LO:

- **L'autonomia:** il discente può utilizzare i LO per acquisire conoscenze e competenze in modo personale, cioè seguendo i dettami dei suoi bisogni e dei suoi tempi di apprendimento;
- **La specificità e l'eshaustività:** un LO deve essere un'unità completa che consenta di apprendere uno specifico contenuto didattico in maniera esaustiva;
- **La multimedialità:** è bene che i LO sfruttino varie tecnologie e stimoli in grado di coinvolgere diversi stili di apprendimento;
- **L'interattività:** il LO deve presentare, tra gli altri, contenuti che prevedano l'attiva interazione del fruitore;
- **L'autovalutazione:** è bene che il fruitore abbia a disposizione strumenti, come dei test, per la propria autovalutazione sia durante il processo di apprendimento presentato da un singolo LO che al termine di un percorso che si può articolare attraverso più LO.

Emerge in seguito una prima questione pedagogica relativa alle modalità di applicazione di uno stesso LO a diversi contesti di utilizzo: occorre infatti chiedersi come inserire un LO all'interno di un percorso formativo, affinché esso possa risultare di volta in volta significativo per i suoi fruitori.

Sono state mosse inoltre alcune critiche rispetto alla modalità di apprendimento basata sui LO, in quanto sembra possa risultare individualistica e asociale. Secondo questa prospettiva, un individuo seduto davanti al suo computer praticerebbe una forma di apprendimento che prescinde dal confronto derivante dal riferimento al gruppo dei suoi pari.

Entrambe le questioni sono strettamente connesse all'impostazione pedagogica che si intende adottare nell'operatività didattica che utilizza i LO.

Nel primo caso, sta al formatore il saper garantire un'adeguata strategia didattica, che preveda anche l'utilizzo di LO, in relazione ai reali bisogni formativi degli utenti, eventualmente integrando la presentazione di LO con ulteriori materiali, costituiti anche da lezioni di carattere tradizionale.

Per quanto riguarda la critica di un apprendimento individualistico, un approccio socio-costruttivista applicato all'e-learning può essere una risposta, fornendo come contesto di utilizzo dei LO una comunità di apprendimento, in cui il singolo può migliorare il proprio percorso formativo in relazione ai contributi del gruppo di appartenenza, mediante la partecipazione ad attività comuni, lo scambio di esperienze ed una negoziazione collettiva continua di significati. prevede quindi l'associazione di ogni componente grafico ad un modello a seconda delle sue caratteristiche logiche.

3. L'UTILITÀ DIDATTICA DEI CRUCIVERBA

Come già accennato, lo strumento che verrà presentato nell'ambito di questa tesi sarà un'applicazione informatica in grado di consentire la generazione di cruciverba e chiaramente la loro successiva visualizzazione e risoluzione.

Tale strumento rientra nell'*Iniziativa Lavagne Interattive* precedentemente citata non solo per la valorizzazione della visualizzazione dei cruciverba tramite le lavagne interattive multimediali, ma anche, e soprattutto, per la loro utilità didattica che li rende una risorsa realisticamente utilizzabile in ambito scolastico in svariate materie d'insegnamento e percorsi didattici, con la possibilità di favorire l'apprendimento tramite diverse modalità d'uso.

Innanzitutto vi sono diversi motivi per cui in molti casi si ritiene importante l'utilizzo di giochi nell'insegnamento, ed alcuni di essi sono i seguenti. I giochi infatti:

- rappresentano il modo naturale dello studente più giovane di esplorare il mondo e di apprendere;
- incoraggiano l'uso del linguaggio e la socialità, soprattutto in caso di giochi di gruppo;
- stimolano la fantasia, sono piacevoli e favoriscono la creatività;
- sono motivanti e favoriscono la partecipazione;
- permettono di spezzare i ritmi di una lezione che a volte possono risultare ripetitivi e noiosi;
- consentono in certi casi di introdurre elementi complessi nel loro utilizzo pratico presentandoli da un punto di vista semplificato.

L'uso didattico del cruciverba, spesso circoscritto all'educazione linguistica come strumento di verifica lessicale, in realtà si adatta a qualsiasi tematica (città, eventi storici, nomi propri, sinonimi e contrari, definizioni scientifiche, matematiche e geometriche, ecc.) unitamente alla possibilità di far creare direttamente nuovi schemi agli studenti. È quindi adatto per attività volte all'acquisizione ed al potenziamento delle competenze di letto-scrittura, al consolidamento della capacità di scrittura regolativa, all'uso del dizionario ed all'apprendimento delle modalità di ricerca in rete.

Introdotta negli anni '70 nella didattica delle lingue, il cruciverba ha avuto un impatto straordinario. In genere è presente nei libri di testo come attività di rinforzo, con l'obiettivo di far ripassare e verificare i termini relativi ad un certo ambito lessicale. Più recente è la consuetudine di ricorrere al suo utilizzo durante la fase di riscaldamento (o *warm up*) al fine di esplicitare termini già conosciuti

dagli studenti e presenti nei testi che andranno poi a leggere o ascoltare; in questo modo viene promossa la grammatica dell'anticipazione, atta a facilitare i processi di comprensione.

Il cruciverba tradizionale è costituito da un griglia vuota e da definizioni, vale a dire enunciati che descrivono i termini da inserire in orizzontale o in verticale nella griglia, ed è di fatto questa la tipologia di cruciverba che vengono prodotti tramite l'applicazione descritta in questa tesi. Vi sono tuttavia delle varianti che vale la pena citare, tra cui gli esempi in seguito riportati:

- **Cruciverba con stimolo visivo:** in questa tipologia le definizioni sono espresse da immagini anziché da frasi, e l'utilità consiste, come illustrato da Mario Cardona¹⁴, nella possibilità per lo studente di ricorrere ad una diversa metodologia di ragionamento ed apprendimento tramite l'utilizzo sia del codice visivo-grafemico sia di quello visivo-figurativo;
- **Cruciverba parzialmente completato:** spesso questi cruciverba sono utilizzati nel lavoro in coppia, in cui uno studente possiede la griglia contenente le parole e le definizioni ad esse associate assenti in quella del compagno e viceversa (per esempio tutti gli elementi orizzontali per uno dei due studenti e quelli verticali per l'altro). I due studenti potranno quindi proporre all'altro delle definizioni di cui essi conoscono la risposta ed eventualmente fornire aiuti al compagno. Questa attività induce in genere un'interdipendenza positiva, stimolando l'intelligenza interpersonale e coinvolgendo anche gli studenti che dimostrano poco interesse a risolvere giochi linguistici;
- **Cruciverba completato:** in questo caso allo studente (o al gruppo di studenti) è assegnato il compito di ideare le definizioni relative ai vocaboli già correttamente inseriti nel cruciverba, per poi proporle ai compagni senza l'ausilio della griglia già completata, in modo da consentire loro di risolvere il cruciverba sulla base delle definizioni appena create. Questa modalità è volta a stimolare maggiormente l'abilità di scrittura.

Questi ed altri esempi di utilizzo dei cruciverba in ambito didattico, unitamente all'introduzione della tecnologia delle lavagne interattive multimediali, possono costituire uno strumento di insegnamento estremamente efficace nel cui si unisce l'utile al dilettevole migliorando l'attenzione, la partecipazione ed il coinvolgimento di ogni studente verso il tema didattico affrontato. E questo è il motivo per cui è nata l'idea di implementare un'applicazione in grado di realizzare tale strumento nell'ambito dell'*Iniziativa Lavagne Interattive*.

¹⁴ Mario Cardona, 2001, *Il ruolo della memoria nell'apprendimento delle lingue. Una prospettiva glottodidattica*, Torino, UTET Libreria, pag. 143

4. CONCETTI E TECNOLOGIE UTILIZZATI

4.1. L'interfaccia utente

Con *interfaccia utente* (in inglese UI, User Interface) si intende un insieme di mezzi tramite cui le persone (gli *utenti* in questione) possono interagire con una macchina o un attrezzo complesso (in generale con il *sistema*). Per permettere un'interazione bilaterale, l'interfaccia utente deve fornire mezzi per permettere all'utente di utilizzare e manipolare il sistema (*input*) ed al sistema di produrre gli effetti derivanti dalle azioni operate dall'utente (*output*).

Per fare un esempio basta pensare ad un'automobile: il suo scopo è quello di trasportare persone ed oggetti, ma non è in grado di farlo autonomamente; ha bisogno di un conducente che la utilizzi in modo da farle svolgere la sua funzione. Per permettere ciò alle automobili vengono montate parti come il volante, i pedali, gli specchi retrovisori e gli indicatori del cruscotto, tutti elementi che fanno parte della loro interfaccia utente e con i quali il conducente può interagire per accedere alle funzionalità della vettura oppure ottenere informazioni su di essa.

Spesso ci si riferisce all'interfaccia utente di sistemi meccanici con il termine *interfaccia umano-macchina* (in inglese HMI, Human-Machine Interface). In particolare con questa nomenclatura ci si riferisce a tutto ciò che compone il "livello" che sta tra la macchina vera e propria e l'utente che la utilizza. Gli elementi più comuni che lo compongono sono tali da permettere azioni di controllo da parte dell'utente (come pulsanti, interruttori, mouse) e da comunicare a quest'ultimo delle informazioni in diverse forme (ad esempio monitor e spie comunicano informazioni visive, mentre sirene ed altoparlanti informazioni sonore).

Un importante fattore da tenere in considerazione in fase di modellazione dell'interfaccia utente è l'*usabilità* della stessa. Esso consiste nel fare in modo da rendere la fase di input la più semplice ed immediata da effettuare e la fase di output la più facilmente e rapidamente interpretabile. Per ottenere un buon grado di usabilità bisogna prendere in considerazione la psicologia e fisiologia umana, nonché cercare di rendere efficace, efficiente e soddisfacente il processo di utilizzo del sistema da parte dell'utente.

Nel campo dell'informatica l'interfaccia utente ha una grande importanza essendo il computer una macchina in grado di svolgere una gamma pressoché infinita di operazioni e programmi.

Rendere il calcolatore uno strumento di facile utilizzo in ogni situazione è sempre stato un grande problema ed in passato non vi si è neppure mai prestata l'attenzione necessaria. Al giorno d'oggi al

contrario questa realtà ha assunto un valore di primaria importanza, tanto che l'*interazione umano-computer* (in inglese HCI, Human-Computer Interaction) è largamente studiata al fine di sviluppare l'interfaccia utente ad un livello tale da rendere tale interazione efficace e piacevole.

I fattori da prendere in considerazione sono molti, come le diverse concezioni e i diversi modelli di interazione con il calcolatore di ogni utente ed il continuo sviluppo tecnologico che può rendere disponibili nuove forme di interazione mai prese in considerazione precedentemente.

Allo stesso modo di ogni altra, per interfaccia utente di un computer (o di un programma) si intendono i controlli usati dall'utente per controllare lo stesso (ad esempio i tasti battuti sulla tastiera ed i movimenti del mouse) e le informazioni che il sistema comunica a chi lo utilizza (generalmente di tipo grafico, testuale o sonoro).

Nella storia del computer, sono numerose le interfacce utente che sono state studiate e sviluppate, ma attualmente quella più comune è indubbiamente l'*interfaccia grafica* (in inglese GUI, Graphical User Interface).

4.1.1. L'interfaccia grafica

La caratteristica principale dell'interfaccia grafica è quella di impiegare icone, indicatori e altri elementi grafici insieme a stringhe di testo per rappresentare le informazioni e le azioni a disposizione dell'utente. Per rendere l'interfaccia la più semplice e la più amichevole possibile, spesso gli elementi grafici utilizzati rappresentano una *metafora* di oggetti reali (è il caso della scrivania o desktop, le finestre e i pulsanti), in modo da rendere chiaro il loro utilizzo anche senza un manuale di istruzioni.

Oltre a quelli appena citati, altri elementi ampiamente utilizzati nelle interfacce grafiche sono i menù, le icone ed il puntatore (ovvero un simbolo che l'utente può muovere, tipicamente con il mouse, per selezionare elementi ed azioni da compiere). Tutti questi elementi, comuni a gran parte delle interfacce grafiche esistenti, sono detti *widget* (traducibile dall'inglese all'incirca con "congegno") o anche widget virtuali per differenziarli dalle loro controparti reali.

L'interazione con il sistema si traduce quindi nella manipolazione degli elementi grafici tramite ad esempio l'utilizzo di mouse e puntatore, il che dà il via all'operazione programmata dagli sviluppatori all'interazione con l'elemento selezionato (ad esempio la pressione virtuale di un pulsante).

Esiste un acronimo che spesso viene utilizzato come sinonimo di interfaccia grafica: si tratta di *WIMP* (in inglese Window, Icon, Menu, Pointing device), che sta in realtà ad indicare un

particolare stile di interazione umano-computer caratterizzato dai quattro principali elementi nominati dall'acronimo stesso, e quindi riconducibile allo studio della già citata HCI. Il motivo per cui il paradigma WIMP e l'interfaccia grafica vengono spesso usati come termini per indicare la stessa cosa è proprio da ricercare nella presenza degli elementi grafici dai quali sono entrambi caratterizzati.

Un'altra caratteristica dell'interfaccia grafica è l'uso di modi e modalità diversi.

I *modi* rappresentano comportamenti diversi di uno stesso input a seconda dello stato del sistema e del programma in esecuzione. Ad esempio un click con il mouse su un elemento grafico può produrre un risultato differente dalla medesima operazione effettuata mentre un'altra particolare funzione è attiva. È tuttavia da notare che un uso massiccio di tali modi può portare ad una riduzione della semplicità d'uso del sistema, poiché l'utente dovrebbe tenere a mente i diversi stati in cui il sistema può trovarsi e i vari spostamenti da uno stato all'altro da effettuare per poter eseguire un'operazione.

Per *modalità* invece si intende una particolare via di comunicazione utilizzata per realizzare input e output. Per quanto riguarda l'input un tipico esempio è rappresentato dalla tastiera del computer che consente di inserire del testo, mentre per l'output un esempio è fornito dal monitor che consente di visualizzare elementi grafici e testo, ed un altro dagli altoparlanti che permettono al sistema di produrre suoni. Chiaramente l'interfaccia può ricorrere a più di una modalità contemporaneamente, spesso anche fornendo informazioni ridondanti o permettendo interazioni non strettamente necessarie, lasciando all'utente la libertà di scegliere tra le diverse vie messe a disposizione.

Le *origini* dell'interfaccia grafica risalgono agli anni '60, ed in particolare vengono associate al nome del dottor Douglas C. Engelbart, il quale sviluppò con il suo team all'SRI (Stanford Research Institute) l'NLS (oN-Line System), ovvero un rivoluzionario sistema collaborativo multi-utente. Nella sua realizzazione vennero impiegati elementi innovativi tra cui l'uso del mouse e di un relativo puntatore grafico con il quale era possibile manipolare dei collegamenti ipertestuali.

Questo sistema è stato poi ulteriormente raffinato ed esteso con l'introduzione di veri e propri elementi grafici dai ricercatori della Xerox PARC (Palo Alto Research Company). Si può affermare che il loro computer sperimentale Xerox Alto, sviluppato nei primi anni '70, era dotato di una vera e propria GUI la quale presentava gli elementi principali di un'interfaccia grafica moderna: un insieme di widget (tra cui finestre, menù, icone, check box, radio button) ed un puntatore grafico controllabile tramite mouse in aggiunta all'uso della sola tastiera. Tale interfaccia grafica, chiamata PUI (PARC User Interface), già soddisfaceva il già citato paradigma WIMP, il quale infatti venne

coniato proprio dal team della Xerox PARC. Anche la metafora della scrivania (o desktop) fa la sua prima comparsa in questo sistema.

Per osservare i primi successi commerciali di un prodotto dotato di GUI è però necessario attendere un decennio e la nascita dei sistemi Lisa e, in particolare, Macintosh, entrambi prodotti dalla Apple Computer.

I lavori di casa Apple, così come la maggior parte delle GUI sviluppate negli anni successivi da differenti compagnie e corporazioni, sono pesantemente ispirati all'originaria interfaccia grafica della PARC, tanto che spesso vengono descritte proprio come interfacce di tipo PARC. Le più popolari attualmente sono senza dubbio quelle utilizzate nei sistemi operativi Mac OS X di Apple (in pratica il discendente del Macintosh) e Windows di Microsoft, e quelle basate sul protocollo X Window System (spesso abbreviato a X11 o solo X). Quest'ultimo, sviluppato al MIT (Massachusetts Institute of Technology) nel 1984, viene distribuito come software libero e fornisce l'ambiente di sviluppo ed i componenti che stanno alla base delle interfacce grafiche. Gestisce ad esempio la struttura e il movimento degli elementi grafici e l'interazione tra essi e l'utente attraverso l'uso di tastiera e mouse, ma non pone alcun vincolo allo stile grafico delle interfacce grafiche che possono essere sviluppate partendo dalle basi da esso fornite.

4.1.2. Altri tipi di interfaccia

Quella grafica non è l'unica interfaccia utente esistente nel campo dell'informatica; ve n'è almeno un'altra che va citata per la sua popolarità: si tratta dell'*interfaccia a riga di comando* (in inglese CLI, Command Line Interface).

Essendo nata negli anni '50, prima della GUI, si può affermare che essa è stata la prima vera forma di interazione umano-computer e tutt'ora è sfruttata in molti sistemi, spesso in collaborazione con la sua controparte grafica.

L'interazione prevista da questa interfaccia consiste nella digitazione di serie di parole e simboli che compongono dei comandi che vengono poi interpretati ed eseguiti dal calcolatore. La semplice digitazione di brevi combinazioni di parole e simboli può attivare funzioni anche complesse, rendendo questa interfaccia molto potente (tanto che numerosi utenti esperti la preferiscono rispetto alla GUI), ma essa presenta anche delle evidenti debolezze: tipicamente i comandi e le loro diverse opzioni sono numerosissimi e difficili da memorizzare (in una GUI l'utente normalmente non deve memorizzare nulla, poiché l'azione auto esplicativa di testo e grafica guida la sua interazione con il sistema, rendendola immediata) e la frequenza degli errori commessi è alta, poiché è sufficiente digitare un carattere errato per far fallire l'operazione.

Un'interfaccia grafica è in grado di rendere il sistema maggiormente *user friendly*, “amichevole” dal punto di vista dell'utente, rispetto ad una CLI. Altre differenze tra le due tipologie di interfacce riguardano il numero di modalità di input (usualmente le CLI prevedono l'uso della sola tastiera) e di modi (le CLI non prevedono un ampio sfruttamento di modi diversi come invece fanno le GUI, questo uso è presente solo nella forma della directory corrente dalla quale i comandi sono eseguiti). Le interfacce a riga di comando più popolari sono quelle fornite dalle shell dei sistemi Unix-like (ovvero programmi comuni ai sistemi operativi che rispondono alle specifiche Unix in grado di fornire una CLI e di interpretare i comandi digitati dall'utente) e dal sistema operativo DOS (Disk Operating System) di Microsoft.

4.2. Java

Java è un linguaggio di programmazione sviluppato dalla Sun Microsystems, ed in particolare dal team capitanato da James Gosling, a partire dall'inizio degli anni '90 e rilasciato nel 1995.

Gli obiettivi usati dagli sviluppatori come punto di riferimento nella sua realizzazione sono i seguenti:

- L'utilizzo della metodologia di programmazione orientata agli oggetti;
- La capacità di eseguire i programmi indipendentemente dalla piattaforma su cui si sta operando;
- Il supporto per il suo utilizzo in reti di computer;
- La capacità di eseguire codice da sorgenti remote in modo sicuro;
- La facilità d'uso, selezionando gli aspetti migliori di altri linguaggi orientati agli oggetti.

Come risultato Java si presenta come un linguaggio di programmazione *orientato agli oggetti*, il che implica la presenza di alcune peculiarità proprie di questa politica di organizzazione del codice, come i basilari concetti di classe, package, oggetto, attributo e metodo e caratteristiche come l'ereditarietà, l'incapsulamento ed il polimorfismo.

Una *classe* è l'unità di base in cui il codice Java è strutturato. Essa definisce in astratto le caratteristiche (*attributi*) ed i comportamenti (*metodi*) di qualcosa (un *oggetto*). È importante che tali componenti rappresentino una metafora sufficientemente coerente con il contesto in cui si sta lavorando, in modo da semplificare la lettura del codice e lo sviluppo dello stesso. Per fare un esempio è possibile implementare una classe di nome Cane la quale va a definire degli *attributi* che caratterizzano tale animale, come ad esempio il colore del pelo, e dei *metodi* che ne rappresentino le capacità, come ad esempio l'abbaiare.

Le classi sono organizzate in una struttura a *package*, dove ognuno di questi ultimi rappresenta un contenitore in cui ad esempio porre le classi della stessa categoria, oppure quelle con funzionalità simili.

Un *oggetto* è un'istanza della classe a cui appartiene, acquisisce i metodi e gli attributi in essa definiti e per ognuno di questi ultimi possiede un valore associato. Ogni oggetto è caratterizzato da uno *stato* che è determinato dall'insieme degli attributi e dei valori da essi presentati. Per fare un esempio l'oggetto di nome Lassie è un Cane, quindi ha la capacità di abbaiare ed il colore del suo pelo acquista il valore “marrone e bianco”.

Il concetto di *ereditarietà* indica la possibilità per le classi Java di ereditare i tratti di un'altra classe, la quale diventa loro “padre”. È possibile ad esempio creare le classi Collie e Chihuahua ed assegnare loro la classe Cane come padre; in questo modo le due nuove classi saranno caratterizzate dagli stessi metodi ed attributi del loro genitore. Nell'ambito di queste due nuove classi figlie è poi possibile aggiungere nuovi attributi e metodi (per esempio assegnando alla sola classe Chihuahua la capacità di tremare), inoltre c'è la possibilità di ridefinire i metodi descritti in Cane (modificando la capacità di abbaiare della classe Chihuahua, descrivendone il suono come molto più squillante).

L'*incapsulamento* fornisce la possibilità di rendere o meno accessibili ad un oggetto i metodi e gli attributi di una classe. Ad esempio il metodo “abbaiare” della classe Cane può necessitare dell'attivazione di altri due metodi come “inspirare” ed “espirare” durante la sua esecuzione, ma ciò non interessa ad un oggetto che richiede l'esecuzione dell'azione “abbaiare”. Ecco che quindi non vi è alcuna utilità nel rendere accessibili tali due metodi. L'utilità dell'incapsulamento è quella di rendere gli oggetti che richiedono l'esecuzione di un metodo come “abbaiare” indipendenti da parti di codice elaborate (in questo caso i metodi “inspirare” ed “espirare”) e probabilmente soggette a modifiche e revisioni future, le quali potrebbero risolversi in conseguenti modifiche anche degli oggetti richiedenti, rendendo disponibile solo una ristretta interfaccia composta da metodi tipicamente semplici.

Il *polimorfismo* descrive la possibilità di un metodo comune a diversi oggetti di comportarsi in maniera differente a seconda della classe da cui l'oggetto deriva. Ad esempio, come già detto, le classi Collie e Chihuahua ereditano il metodo “abbaiare” dalla classe padre Cane, ma nella seconda tale metodo è stato rielaborato in modo da ottenere un suono più squillante, sicché l'oggetto Lassie, istanziato dalla classe Collie, e l'oggetto Fido, istanziato dalla classe Chihuahua, abbaieranno in maniera diversa. Un'altra tipologia di polimorfismo è il diverso comportamento di un operatore (come ad esempio il segno “+”) all'interno del codice a seconda del tipo di dati su cui si sta operando. Ecco che quindi può per esempio sommare due numeri interi, ma anche concatenare due stringhe.

L'indipendenza dalla piattaforma (hardware e software di cui è dotato il calcolatore) in cui si opera dei programmi sviluppati in Java, e quindi la *portabilità*, è ottenuta tramite l'introduzione di un particolare software, la *macchina virtuale*, e di un particolare stato intermedio di compilazione del codice, detto bytecode.

A differenza di altri linguaggi di programmazione, quando si compila un programma scritto in Java il codice non viene tradotto nel linguaggio nativo della piattaforma in uso, e quindi pronto per l'esecuzione da parte del calcolatore al quale si lavora, ma bensì nel più semplice bytecode, indipendente dall'ambiente in cui si sta lavorando. Tale codice viene successivamente eseguito dalla macchina virtuale, indipendentemente da dove essa si trovi e da dove tale bytecode provenga. Questo sistema, per quanto ottimale dal punto di vista della portabilità, è stato criticato per la sua lentezza se confrontato con quello tradizionale di altri linguaggi come il C, ma attualmente la macchina virtuale è stata resa maggiormente sofisticata ed è in grado di applicare delle tecniche per rendere il processo più performante, come ad esempio la traduzione del bytecode nel linguaggio nativo della piattaforma in uso al momento dell'esecuzione del programma. La Sun Microsystems rende ufficialmente disponibile la macchina virtuale Java per sistemi operativi Microsoft Windows, Linux e Solaris.

Un'altra importante caratteristica di Java è il *garbage collection* (il cui acronimo è GB), ovvero una forma di gestione della memoria, completamente automatico.

In molti linguaggi di programmazione è necessario per i programmatori allocare nella memoria del calcolatore lo spazio richiesto dagli oggetti creati dai loro programmi e nella stessa maniera anche deallocare tale spazio nel momento in cui esso non è più utile. Se un programmatore dimentica di svolgere quest'ultima operazione, il risultato è una certa quantità di memoria occupata inutilmente ed inutilizzabile (si tratta di un *memory leak*); se poi si richiede di deallocare per errore dello spazio non precedentemente allocato, allora le conseguenze sono imprevedibili e potrebbero portare ad un crash del programma.

In Java il programmatore non deve più preoccuparsi di queste problematiche poiché il garbage collection automatico si occupa di deallocare lo spazio in memoria quando questo non è più utile. Ciò è possibile tenendo traccia del ciclo vitale degli oggetti creati; nel momento in cui il programma in esecuzione e tutti gli altri oggetti non possiedono più alcun riferimento ad un oggetto, ciò significa che esso è divenuto inutile, quindi il sistema dealloca lo spazio da esso occupato.

Le distribuzioni di Java forniscono al programmatore numerose funzionalità già esistenti evitandogli la fatica di implementare ogni singola operazione, anche la più elementare, tramite le *API* (acronimo di Application Programming Interface). Si tratta dell'interfaccia (ovvero i metodi accessibili) fornita da classi raggruppate in package fornite con il pacchetto Java di Sun Microsystems.

Le funzionalità implementate da tali classi sono numerose (possono variare a seconda della distribuzione che si possiede) e coinvolgono aspetti come la gestione delle strutture dati, l'input e l'output, il supporto per le reti, la sicurezza e l'interfaccia grafica.

Oltre alle applicazioni normali, Java fornisce gli strumenti per svilupparne anche alcune di tipo particolare: è il caso delle applet, delle servlet e delle JavaServer Page (JSP).

Le *applet* sono delle applicazioni Java contenute in altre applicazioni, tipicamente una pagina web visualizzata da un browser; le *servlet* sono invece applicazioni "lato server", ovvero in grado di generare risposte (tipicamente delle pagine HTML) a delle richieste provenienti da client (tipicamente richieste HTTP); le *JavaServer Page* si presentano come pagine HTML contenenti blocchi di codice Java. Esse sono compilate in servlet la prima volta che vi viene effettuato l'accesso, dopodiché è tale servlet a rispondere alle richieste future.

Il linguaggio di programmazione Java mette a disposizione un ambiente per la creazione di interfacce grafiche per le proprie applicazioni, indipendenti dalla piattaforma su cui si sta lavorando. Si tratta delle *JFC* (ovvero Java Foundation Classes) che presentano le seguenti caratteristiche:

- **Componenti della GUI:** ne è inclusa un'ampia gamma, che va da pulsanti e check box a tabelle e testi, fornendo il supporto di specifiche funzionalità a seconda dei componenti stessi, con la quale è possibile costruire interfacce grafiche complete e funzionali;
- **Supporto per look-and-feel addizionali:** per look-and-feel si intende il motivo grafico che decora i componenti della GUI. Le JFC ne mettono a disposizione diversi tra cui quello che caratterizza il sistema operativo Microsoft Windows ed inoltre forniscono il supporto per l'utilizzo di altri personalizzati;
- **API per l'accessibilità:** le JFC includono il supporto per tecnologie che forniscono assistenza per l'utente nel ricavare informazioni dall'interfaccia, come ad esempio display Braille (che riproducono in alfabeto Braille ciò che appare sullo schermo) e screen reader (che trasformano in voce il testo presente sullo schermo) per non vedenti;

- **API Java 2D:** consente agli sviluppatori di incorporare elementi grafici bidimensionali, testi, animazioni ed immagini nelle loro applicazioni. Gli stessi componenti grafici della GUI sono costruiti sulla base di questa API;
- **Internazionalizzazione:** gli sviluppatori possono creare applicazioni in grado di interagire con utenti di qualsiasi cultura e linguaggio, ovvero che sono in grado di accettare testo scritto in svariati linguaggi, anche se questi prevedono l'uso di migliaia di caratteri diversi e di convenzioni particolari come la scrittura da destra a sinistra.

Inizialmente tale ambiente di sviluppo per le GUI era rappresentato esclusivamente da *AWT* (acronimo di Abstract Window Toolkit), il quale mette a disposizione una grande quantità di componenti grafici e le loro funzionalità, ma si rifà all'architettura nativa della piattaforma in cui si opera per realizzarli ed in tal modo non è possibile personalizzarne il motivo grafico.

Dal 1997 invece il toolkit *Swing* si è affiancato ad *AWT* fornendo non solo un maggior numero di elementi grafici disponibili, ma anche la capacità di disegnarli (utilizzando comunque nel fare ciò delle funzionalità di *AWT*) senza appoggiarsi al sistema sottostante e quindi un maggior livello di personalizzazione che si risolve nella lista di caratteristiche elencate in precedenza.

AWT rimane comunque il toolkit che sta alla base del funzionamento dell'interfaccia grafica gestendone ad esempio gli eventi (come la pressione di un pulsante tramite il mouse) e l'interazione con dispositivi di input (come la tastiera ed il mouse).

L'obiettivo generale del progetto *Swing* era quello di costruire un'ampia gamma di componenti della GUI per permettere agli sviluppatori di implementare più rapidamente delle potenti interfacce nelle loro applicazioni. Il team di sviluppo di *Swing* lavorò quindi tenendo in considerazione delle linee guida che avrebbero condotto a tale obiettivo; esse prevedevano che *Swing* dovesse:

- Essere interamente implementato in Java, in modo da promuovere la portabilità e facilitare la mantenibilità;
- Fornire una singola API in grado di supportare diversi look-and-feel;
- Sfruttare la potenza della programmazione model-driven fornendo un'interfaccia ad alto livello di astrazione, senza doverla richiedere a delle API di più alto livello;
- Aderire ai principi dell'architettura *JavaBeans*, in modo da assicurare un corretto funzionamento dei componenti grafici anche in ambienti come le IDE (acronimo di Integrated Development Environment) ed altri strumenti di sviluppo;
- Essere compatibile con le API *AWT* in modo da poterle sfruttare al meglio.

L'architettura di Swing affonda le sue radici nel design denominato model-view-controller (MVC) (Fig.1), il quale prevede la separazione di un'applicazione visiva in tre parti separate:

- Il *model* (modello): che rappresenta i dati dell'applicazione;
- La *view* (vista): che è la rappresentazione visiva dei dati;
- Il *controller* (controllore): che ha il compito di raccogliere gli input forniti dall'utente nella view e di tradurli in modifiche al model.

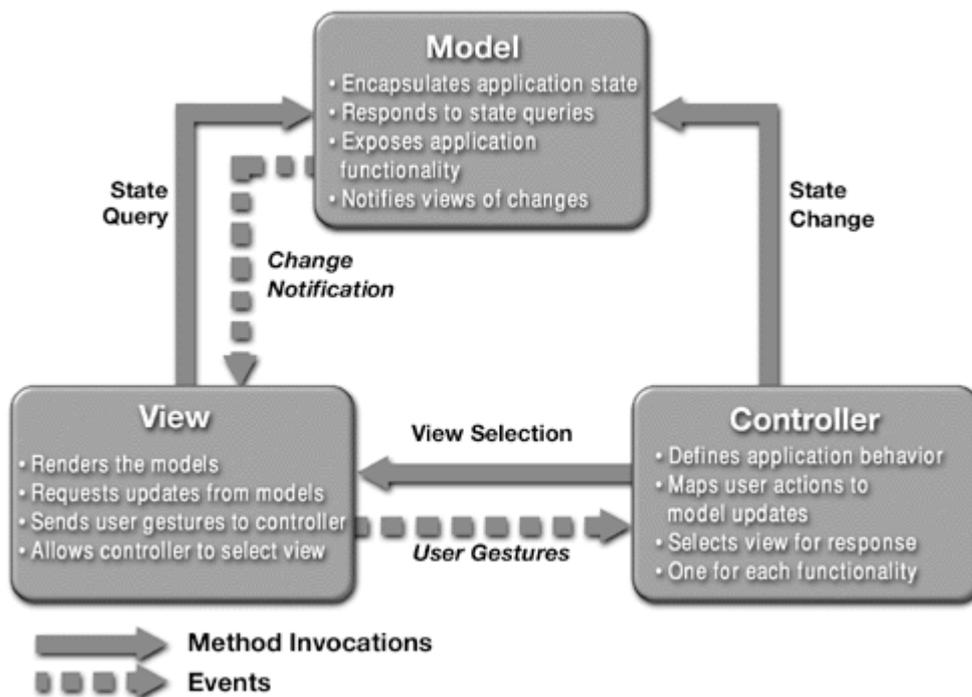


Fig.1: Rappresentazione del pattern architetturale denominato model-view-controller (MVC)

Si è però reso evidente, durante lo sviluppo di Swing, che tale design non poteva funzionare al meglio in pratica, poiché le due parti di view e controller richiedevano una stretta collaborazione (ad esempio era estremamente difficile implementare un controller generico che non sapesse nulla riguardo alle specifiche della view). È stata pertanto fatta la scelta di fondere tali due entità in una sola, spesso chiamata *UI delegate* (ovvero delegato dell'interfaccia grafica). Questo design, derivato da una modifica di quello MVC, è spesso chiamato *separable model architecture*. Tale architettura prevede quindi l'associazione di ogni componente grafico ad un modello a seconda delle sue caratteristiche logiche.

4.3. Flash

Adobe Flash (o più semplicemente Flash) è una tecnologia di orientamento prevalentemente grafico che conta tra le sue funzionalità il supporto per la grafica vettoriale e quella raster, un linguaggio di scripting chiamato ActionScript e lo streaming bidirezionale di audio e video. Con il termine Flash ci si riferisce anche a due particolari applicazioni:

- **Adobe Flash Professional:** ovvero un ambiente di sviluppo con il quale costruire oggetti Flash;
- **Adobe Flash Player:** ovvero una macchina virtuale in grado di visualizzare gli oggetti Flash.

La tecnologia Flash è stata lanciata nel 1996 da Macromedia (ora parte di Adobe Systems) come evoluzione del progetto FutureSplash della compagnia FutureWave Software (allora acquistata da Macromedia), ed è divenuta il mezzo principale con cui aggiungere elementi grafici interattivi nelle pagine web; con essa è infatti possibile creare animazioni, banner pubblicitari, vari componenti per pagine web, e, ultimamente, interi siti web, giochi ed applicazioni. Gli elementi principali che possono essere contenuti in un oggetto Flash sono i seguenti:

- **Forme vettoriali:** sono gli oggetti principali con cui Flash permette di lavorare;
- **Testo:** può essere sia statico che dinamico, inoltre sono previste caselle per l'input di testo;
- **Immagini raster:** Bitmap, GIF, Jpeg, PNG, TIFF sono alcuni dei formati supportati;
- **Audio:** sono supportati formati come MP3, WAV ed altri che possono essere eseguiti sia in streaming che per singoli effetti sonori;
- **Video:** con il supporto di formati come AVI, QuickTime, MPEG, Windows Media Video e FLV;
- **Altre animazioni Flash.**

I file Flash, tradizionalmente chiamati “filmati Flash”, hanno l'estensione .swf e tipicamente rappresentano degli oggetti inclusi in pagine web che vengono visualizzati tramite il Flash Player, il quale può essere sia stand alone che integrato nel browser che si usa per visualizzare le pagine web, infatti attualmente la maggior parte dei browser moderni lo includono nativamente.

Se inizialmente Flash si basava quasi esclusivamente sulla grafica e sulle animazioni, ora un altro elemento accompagna tali contenuti: si tratta dell'interattività, divenuta ormai un fattore di primaria

importanza nella realtà della tecnologia Flash ed ottenuta grazie al linguaggio di scripting interno denominato *ActionScript*, un linguaggio orientato agli oggetti giunto ormai alla sua terza versione (risalente al 2006). Tale linguaggio è basato sullo standard ECMAScript, le cui specifiche sono state pubblicate da Ecma International nel 1997¹⁵, ed è quindi piuttosto simile a JavaScript, rientrando anch'esso nello stesso standard.

Tramite ActionScript è possibile applicare dei particolari comportamenti agli oggetti ed ai fotogrammi di cui le animazioni Flash sono composte; grazie all'evoluzione che lo ha interessato nel corso degli anni, se inizialmente erano realizzabili solo poche e semplici azioni, è ora possibile interagire con le applicazioni Flash tramite pulsanti, menù, liste selezionabili, campi di testo, ecc., fino ad ottenere intere applicazioni interattive complesse, come le RIA (acronimo di Rich Internet Application), con performance di esecuzione sempre maggiori.

Il Flash Player non è solo uno strumento di visualizzazione, ma contiene anche una macchina virtuale per l'esecuzione del codice ActionScript e quindi per rendere possibile anche l'interazione tra l'utente e le applicazioni Flash. Più in dettaglio, l'ultima versione del Player, la nona, incorpora ben due macchine virtuali, una per il codice ActionScript delle versioni 1 e 2, ed una per il codice della terza versione di ActionScript, necessaria vista la completa revisione dell'architettura del linguaggio.

Un'interessante caratteristica di Flash è la possibilità, durante lo sviluppo di un'applicazione, di mantenere in una libreria gli elementi utilizzabili più volte nella stessa, i quali vengono chiamati simboli. Esistono diverse tipologie di simboli, fra le quali si contano semplici oggetti grafici, filmati, pulsanti, immagini bitmap, ecc. I vantaggi che questa caratteristica introduce sono notevoli, in quanto essa semplifica ad esempio il lavoro dello sviluppatore, il quale può modificare tutte le istanze di un simbolo presenti nell'applicazione con una singola azione, cioè modificandone il relativo simbolo in libreria. Essa consente inoltre, grazie a questo paradigma che si può accostare a quello orientato agli oggetti di un linguaggio di programmazione come ad esempio Java, di ridurre le dimensioni del file Flash .swf prodotto.

Come già detto, il file finale prodotto dallo sviluppo di un'applicazione Flash assume l'estensione .swf e viene visualizzato tramite il Flash Player. Esistono tuttavia dei software *decompilatori* in grado di risalire facilmente al codice sorgente ed alla struttura di tali applicazioni .swf e ciò pone un problema per gli autori che desiderano proteggere il contenuto dei propri prodotti. Per risolvere tale

¹⁵ Le specifiche del linguaggio ECMAScript si trovano nello standard ECMA-262, giunto alla sua terza edizione (datata 1999) e disponibile al link <http://www.ecma-international.org/publications/files/ecma-st/ECMA-262.pdf>

problematica sono stati sviluppate delle applicazioni definite *offuscatori* con il compito di impedire ai decompilatori di essere in grado di ricostruire i dettagli dell'implementazione delle applicazioni Flash. Il risultato è che i software decompilatori spesso falliscono durante l'analisi del file .swf; tuttavia non esiste ancora un offuscatore in grado di proteggere l'intero contenuto di un'applicazione Flash.

Il formato Flash è al giorno d'oggi estremamente diffuso nel mondo, tanto da rappresentare ormai quasi uno standard per la creazione di contenuti animati ed interattivi. In un'indagine statistica svolta dalla Millward Brown su commissione di Adobe Systems, è stato rilevato che nei paesi caratterizzati da un mercato maturo nel giugno 2007 più del 99% dei computer in grado di navigare in Internet erano in grado di visualizzare un'applicazione Flash in quando avevano montato una versione (non necessariamente la più recente) di Flash Player¹⁶.

La tecnologia Flash è ufficialmente compatibile con le piattaforme dotate di sistemi operativi Windows, Macintosh, Linux e Solaris. Inoltre negli ultimi anni Flash sta cercando di conquistare anche altre piattaforme che vanno oltre il solo computer, come la telefonia e le console.

È doveroso annotare anche alcune note negative che caratterizzano la tecnologia Flash: innanzitutto, come conseguenza alla sua enorme diffusione, i contenuti Flash sono diventati l'ordine del giorno nelle pagine web e spesso essi consistono in banner pubblicitari che possono essere anche aggressivi ed invadenti viste le loro potenzialità grafiche e sonore. Ciò ha portato alla necessaria nascita di numerosi strumenti per il blocco dei contenuti Flash, spesso contenuti nativamente nei browser moderni alla pari del Flash Player, dando la possibilità all'utente di scegliere se visualizzare o meno tali applicazioni. Un'altra problematica che però tende ad avere un impatto sempre più ridotto col passare del tempo è l'impossibilità (a causa dell'assenza del Flash Player) o la difficoltà (a causa della connessione ad Internet lenta) per gli utenti di accedere ai contenuti Flash non disponibili in altri formati più facilmente raggiungibili.

¹⁶ Per maggiori informazioni visitare il link http://www.adobe.com/products/player_census/flashplayer/

5. L'APPLICAZIONE

L'applicazione presa in esame in questa tesi, da me sviluppata nel contesto dell'*Iniziativa Lavagne Interattive*, consiste in uno strumento per la generazione ed il successivo utilizzo di cruciverba interattivi. Tale strumento è stato ideato nell'ambito dell'Iniziativa durante gli incontri tra gli attori partecipanti al progetto SLIM4DIDA.

I requisiti basilari che devono caratterizzare l'applicazione sono stati quindi formalizzati dal professor Marco Ronchetti e si possono riassumere nei seguenti punti:

- L'applicazione deve essere in grado di consentire agli insegnanti di costruire cruciverba;
- La parte dell'applicazione con il compito di generare i cruciverba deve risultare di facile utilizzo e di gradevole aspetto, inoltre deve consentire il maggior controllo e la maggior personalizzazione possibile all'insegnante, senza per questo generare confusione;
- L'applicazione deve essere in grado di generare dei learning object sotto forma di cruciverba interattivi;
- I cruciverba prodotti dall'applicazione devono presentarsi di aspetto gradevole e devono poter essere utilizzati facilmente, in modo da permettere all'utente di controllare il gioco in maniera semplice, intuitiva e funzionale;
- Sia l'applicazione generatrice di cruciverba che i cruciverba da essa prodotti devono poter essere utilizzati sul maggior numero di piattaforme differenti senza problemi.

Sulle basi di tali requisiti si è cercato di individuare le tecnologie ideali da utilizzare nello sviluppo dell'applicazione. La scelta per quanto riguarda il prodotto finale, ovvero i cruciverba, è ricaduta sulla tecnologia Flash vista la sua ampia diffusione sulla maggior parte dei calcolatori e le sue potenzialità dal punto di vista grafico, mentre quella per il linguaggio di programmazione con cui implementare l'applicazione generatrice di cruciverba è subito ricaduta su Java dato il suo supporto alla portabilità e la relativa semplicità nel realizzare applicazioni funzionali dotate anche di una buona interfaccia grafica.

Decise in via definitiva le due tecnologie Java e Flash, il passo successivo consisteva nell'affrontare il passaggio dall'una all'altra, ovvero la domanda da porsi era: come fare in modo che l'applicazione Java produca come output un oggetto Flash che implementi un cruciverba, possibilmente incluso in una pagina web e pronto per l'uso immediato? Tale risultato non è immediatamente realizzabile vista l'impossibilità di creare oggetti Flash direttamente da

un'applicazione Java. Ciò ha reso evidente la necessità di ricorrere ad una tecnologia intermedia che possa essere facilmente prodotta come output dall'applicazione Java e quindi convertita in un oggetto Flash pronto alla visualizzazione.

È da notare che la potenza del linguaggio di scripting di Flash, ActionScript, supporta pienamente la lettura di dati da documenti in formato XML e che questi ultimi sono facilmente ottenuti come output da un'applicazione Java visto che si tratta fondamentalmente di testo formattato secondo una particolare sintassi; ecco che quindi proprio XML si candida come possibile tecnologia intermedia tra generatore di cruciverba in Java e cruciverba vero e proprio in Flash.

Riassumendo la situazione, ora si sarebbe in grado di produrre un documento XML nel quale si inserirebbero tutti i dati necessari alla creazione di un cruciverba, ma resta il fatto che dall'altra parte sarebbe necessario un oggetto Flash pronto a leggere il documento XML prodotto ed a mostrare il cruciverba basandosi interamente su tale documento. In altre parole, visto che l'applicazione Java è in grado di modellare solo il documento XML, è necessario che esso contenga le informazioni riguardo a tutti gli aspetti dell'oggetto Flash che descrive, a partire dai dettagli della struttura grafica di quest'ultimo fino ai vari controlli per la realizzazione dell'interattività, e di conseguenza dev'essere approntato un oggetto Flash praticamente vuoto, contenente esclusivamente il codice ActionScript in grado di leggere e di interpretare tutti i dati contenuti nel documento XML che gli viene consegnato dall'applicazione Java.

La struttura dell'intero progetto inizia a prendere forma: ora è prevista l'implementazione dell'applicazione Java che permette la creazione di cruciverba e che immagazzina i dati relativi ad ognuno di essi in un documento XML da una parte, mentre dall'altra di uno speciale oggetto Flash in grado di leggere tali documenti XML e di riprodurre graficamente e funzionalmente i dati in essi contenuti.

5.1. Gaupe

A questo punto della progettazione, un'applicazione liberamente utilizzabile si è dimostrata estremamente utile in quanto le funzionalità da essa implementate sono esattamente quelle richieste dal progetto in corso: si tratta dell'applicazione chiamata *Gaupe*, prodotta dalla compagnia norvegese Svevid¹⁷ nel 2006, il cui nome significa "lince" in norvegese, animale da cui eredita anche il suo logo.

Gaupe è una soluzione gratuita per lo sviluppo di applicazioni web dotate di un'interfaccia ricca, basata su Flash e generata dinamicamente senza il bisogno di altro che di un editor di testi; questa è

¹⁷ Vedere la home page della compagnia: <http://www.svevid.com/>

la definizione dell'applicazione ideata direttamente dagli sviluppatori, i quali ne hanno ideata una ancora più breve e diretta in grado di spiegare meglio l'utilità di Gaupe nell'ambito del progetto in esame: si tratta di un'applicazione Flash che carica del codice XML e lo trasforma a sua volta in un'applicazione Flash. Esattamente ciò che occorre al progetto in esame.

Gaupe è in grado di leggere dal documento XML che gli si comunica elementi come istanze di componenti grafici, sintassi CSS (acronimo di Cascading Style Sheets, o più semplicemente “fogli di stile”) per descrivere come tali componenti devono apparire graficamente e codice di scripting che verrà poi interpretato. Ovviamente tale file XML che viene elaborato da Gaupe deve essere strutturato in maniera tale da essere compreso dall'applicazione stessa. L'applicazione è composta da tre elementi principali:

- **La libreria dei componenti:** contiene una serie di componenti grafici Flash dei quali si possono visualizzare delle istanze definendole nel documento XML;
- **L'interprete di script:** è scritto in ActionScript ed è in grado di interpretare gli script letti dal documento XML in un linguaggio che, pur presentando delle differenze sintattiche, ricorda proprio lo stesso ActionScript. La potenza di questo linguaggio di scripting è inferiore a quella di ActionScript, rende infatti disponibile solo una parte delle sue funzionalità;
- **Un insieme di funzioni e classi:** è possibile eseguire diverse azioni grazie ad esse, come ad esempio ridisegnare graficamente gli elementi grafici, interpretare il codice CSS e svolgere comunicazioni con paradigma client-server.

Tutto ciò è contenuto in un oggetto Flash chiamato gaupe.swf. Oltre ad esso l'applicazione nel suo complesso è composta anche da una pagina web, modificabile a piacere, con già incluso in essa lo script (in linguaggio JavaScript) necessario per visualizzare l'oggetto Flash in essa e per comunicare ad esso il documento XML da leggere. In caso di assenza del Flash Player viene mostrato il link per il download di tale strumento.

L'oggetto Flash che viene richiamato dalla pagina web non è quello principale di cui si è parlato poco sopra, ma un altro (con nome client.swf) con il compito di visualizzarne uno chiamato loader.swf, il quale mostra un'animazione di attesa mentre quello principale carica il codice XML e lo interpreta, prima di aprire l'applicazione principale, ovvero l'oggetto gaupe.swf. Gli sviluppatori di Gaupe mettono a disposizione il codice sorgente dell'oggetto Flash loader.swf che quindi può essere modificato a piacimento.

Al momento del download dell'applicazione è possibile personalizzare la stessa tramite la scelta di alcune opzioni:

- La possibilità di ridurre le dimensioni (già assai ridotte) dell'applicazione selezionando manualmente, al momento del download, i componenti grafici che si vogliono inserire nella libreria dei componenti; essi però non potranno ovviamente essere usati;
- La possibilità di applicare una protezione all'applicazione: essa si traduce nella scelta dell'URL in cui Gaupe si troverà, l'unica locazione in cui essa funzionerà (se qualcuno dovesse copiare l'applicazione e tentare di eseguirla altrove, essa non funzionerà);
- La possibilità di avere incluso nel download anche un file scritto in linguaggio PHP in grado di trasferire dati tra Gaupe e degli script dal lato server. Scegliendo questa opzione l'applicazione non funzionerà in modalità offline;
- La retro compatibilità con versioni vecchie di Flash: attualmente Gaupe dà la possibilità di essere compatibile con la versione 8 di Flash (ma non con le precedenti), oppure di essere compatibile con tutte le versioni anche precedenti, ma senza sfruttare le funzionalità aggiuntive introdotte dalla versione 8. Ad oggi non è ancora supportata la versione 9 della tecnologia Flash.

Nel documento XML i vari componenti grafici che vanno a costituire l'applicazione sono rappresentati da tag caratterizzati dalla lettera iniziale maiuscola (vi sono ad esempio i tag <Text> per campi di testo e <Movie> per le animazioni), ognuno dei quali può essere descritto da degli attributi, i più comuni tra i quali sono quelli che danno informazioni circa la posizione che il componente dovrà avere (chiamati x ed y), le dimensioni che presenterà (*height* e *width*) ed il suo identificatore univoco (*id*). Quest'ultimo attributo può assumere come valore qualsiasi sequenza di lettere e numeri che dev'essere diversa da tutte quelle degli altri identificatori nell'ambito dell'intera applicazione. La sua utilità è quella di fare riferimento ad esso nel caso si vogliano applicare degli script ad una particolare istanza di un componente.

I componenti che vanno a comporre il documento XML seguono una struttura ad albero, in modo tale che un componente ne possa contenere altri a sua volta, generando una relazione di parentela padre-figli. Se un componente padre viene ridimensionato o spostato, tali modifiche vengono ereditate da tutti i suoi componenti figli. Il documento XML deve presentare un componente di tipo <Content> come nodo radice di tale struttura ad albero.

È prevista la presenza di un particolare tag figlio del componente radice, ma privo di qualsiasi altra parentela (non può contenere altri componenti a sua volta): si tratta del tag <interpret>. Il suo

interno è il luogo dove è possibile digitare l'intero codice di scripting che verrà poi interpretato da Gaupe; è quindi possibile creare ed utilizzare variabili, scrivere delle funzioni ed anche utilizzare parte delle classi messe a disposizione da ActionScript.

Un altro tag particolare è quello chiamato *<handlers>*, ogni istanza del quale deve essere figlia di un componente come ad esempio un pulsante. La sua funzione è quella di inizializzare la gestione di determinati eventi che avvengono sul componente padre (ognuno di tali eventi identificato da un attributo) lanciando una funzione specificata per ognuno di essi (il valore associato all'attributo che identifica l'evento interessato), la quale deve essere presente nel codice all'interno del tag *<interpret>*.

Infine, un ultimo tag speciale è quello chiamato *<style>* il quale, similmente ad *<interpret>*, è figlio del componente radice ma privo di figli e contiene in sé tutto il codice CSS che descrive le caratteristiche grafiche dei componenti usati nell'applicazione.

Lo sviluppo dell'applicazione è quindi proseguito basandosi pienamente sulle potenzialità dell'applicazione Gaupe.

5.2. Il cruciverba in Flash

Il passo successivo è stato quello di costruire un documento XML da passare a Gaupe il quale contenga tutti i dati per visualizzare un cruciverba perfettamente funzionale e pronto per essere utilizzato, in modo da progettare la struttura dello stesso file XML che la futura applicazione Java dovrà restituire come output una volta ultimata la creazione di un cruciverba tramite essa.

Innanzitutto sono stati delineati ancora una volta i requisiti che il cruciverba ultimato avrebbe dovuto presentare:

- La visualizzazione della griglia di caselle che compongono il cruciverba;
- La possibilità di visualizzare le singole caselle in modo normale, annerito, oppure contenenti il numero che identifica le definizioni delle parole che hanno origine dalla casella interessata;
- La possibilità di scrivere una lettera in ogni casella con l'esclusione di quelle annerite;
- La visualizzazione di tutte le definizioni del cruciverba o in alternativa solo quelle che interessano la casella selezionata al momento;
- La presenza di controlli in grado di svolgere alcune utili operazioni (sono inizialmente state individuate le funzioni di cancellazione di tutte le lettere digitate nel cruciverba, di verifica

dello stato attuale del cruciverba con la notifica delle lettere scorrette, di visualizzazione della soluzione completa del cruciverba).

Il primo requisito soddisfatto è stato quello che prevede il disegno della griglia che va a formare il cruciverba. Ogni singola casella è stata costruita sfruttando due componenti tra quelli messi a disposizione da Gaupe: le caselle di testo (caratterizzate dal tag `<Text>`) e le caselle per l'input di testo (indicate con il tag `<EditableText>`). In pratica viene disegnata una casella di testo per l'input di forma quadrata e dotata di un sottile bordo nero, all'interno della quale l'utente potrà digitare una singola lettera; essa quindi va a rappresentare una singola casella nel disegno complessivo del cruciverba. Nell'angolo alto a sinistra di ognuna di tali caselle ne viene inserita un'altra più piccola e non atta a prendere in ingresso input dall'utente con lo scopo di riportare in essa il numero che identifica le definizioni associate alla casella interessata, mentre chiaramente rimarrà vuota nel caso non vi sia alcuna parola che abbia origine in tale casella.

Per trovare le coordinate di ogni casella basta semplicemente sommare agli attributi x ed y dei campi di testo il valore del lato di una casella via via che si aumentano le righe e le colonne. Se quindi la prima casella in alto a sinistra avrà 0 come valore sia di x che di y , quella adiacente alla sua destra avrà lo stesso valore di y , ma x sarà incrementata di tanto quanto la larghezza di una casella. Lo stesso dicasi per quella adiacente sottostante, ma in questo caso x rimarrà invariata ed y sarà la coordinata da incrementare di tanto quanto l'altezza di una casella.

Fatto ciò, successivamente è stata ideata una soluzione per raffigurare le caselle annerite: nel codice XML è possibile inserire anche quelli che poi saranno interpretati come “simboli” Flash da Gaupe, ovvero un oggetto grafico del quale posso richiamare quante istanze voglio nell'applicazione Flash finale. Così ne è stato realizzato uno (tramite il tag `<Drawing>`) costituito da un quadrato nero lievemente più piccolo di una casella del cruciverba circondato da un sottile bordo identico a quello delle altre caselle.

Ora, le caselle annerite non vengono realizzate tramite due campi di testo, uno normale ed uno per l'input, come per quelle normali, ma bensì da uno singolo normale vuoto e privo di bordo, con l'unico scopo di fare da padre ad un'istanza del “simbolo” creato per raffigurare la casella nera. L'istanza è richiamata all'interno del tag `<Text>` dal tag `<draw>` che richiede di specificare un attributo chiamato *id* il quale deve essere lo stesso di quello scelto per il “simbolo” in questione. Tale istanza, essendo quindi figlia della casella di testo in cui si trova, ne eredita le coordinate spaziali alle quali si sommano quelle relative che descrivono il “simbolo”.

Un'altra operazione da fare in questa fase di disegno della griglia del cruciverba è stata quella di permettere all'utente di digitare una sola lettera all'interno di ogni casella. Questa restrizione è

applicabile semplicemente grazie ad un attributo che può essere usato per descrivere il tag `<EditableText>`: si tratta di *maxLength* che deve assumere come valore il numero massimo di caratteri che possono essere digitati nella casella di testo relativa, e che in questo caso è quindi posto ad 1.

L'ultima operazione svolta riguardante le caselle del cruciverba è stata l'implementazione di codice CSS per la formattazione del testo usata nella scrittura delle lettere che verranno digitate nelle caselle. Si vuole ad esempio che il carattere sia sufficientemente grande e che la lettera appaia allineata perfettamente al centro della casella. Tutte queste informazioni sono inserite nel tag `<style>` del foglio XML descrivendo l'aspetto del testo di ogni casella per l'input di testo, ovvero le `<EditableText>`.

A questo punto rimane da progettare il design di tutte le informazioni ed i controlli che stanno al di fuori del cruciverba. Come già detto sono stati subito identificati, anche analizzando applicazioni simili già esistenti in rete e valutandone la funzionalità, alcuni elementi che devono entrare a far parte dell'interfaccia del cruciverba in Flash: prima di tutto è necessario visualizzare le definizioni dell'intero cruciverba oppure solo quelle della casella in cui si sta scrivendo al momento; tra le due opzioni è stata scelta la seconda poiché è apparsa una soluzione più leggibile rispetto all'alternativa che prevede di visualizzare l'intero elenco (che può essere anche molto lungo) di definizioni. Ciò è stato facilmente realizzabile tramite l'introduzione di un nuovo campo di testo posto alla destra della griglia del cruciverba con l'intenzione di stamparvi dinamicamente la o le definizioni associate alla casella in cui ci si posiziona per digitare una lettera.

I successivi controlli che si è ritenuto opportuno implementare sono quelli riguardanti la cancellazione di tutte le lettere già digitate nel cruciverba, la verifica dello stato del cruciverba con l'analisi della correttezza delle lettere digitate ed infine la visualizzazione della soluzione completa del cruciverba. Tutti e tre questi controlli sono stati aggiunti all'interfaccia dell'applicazione Flash tramite l'introduzione dei tre relativi pulsanti (identificati dal tag `<Button>` nel documento XML) il primo descritto dal testo "Cancella tutto", il secondo da "Verifica" ed il terzo da "Vedi soluzione", anch'essi posizionati alla destra del cruciverba, dove in pratica si è venuta a formare la prima versione dell'interfaccia di controllo dell'applicazione. Il calcolo delle coordinate alle quali essa viene a trovarsi dipendono dalle dimensioni massime che vengono fissate per il cruciverba (tale scelta è ricaduta su 16 per 12 caselle), in modo da far sì che tutti i controlli vengano a trovarsi a destra dell'intero cruciverba anche nel caso in cui ne venga costruito uno che sfrutti proprio tali dimensioni massime.

Infine è apparso opportuno aggiungere, nuovamente a destra del cruciverba, un'ulteriore campo di testo per la visualizzazione di messaggi che notificano l'avvenuta esecuzione di un evento, come ad esempio la completa cancellazione del cruciverba, oppure rispondano a richieste dell'utente, come il dare informazioni sullo stato attuale del cruciverba con messaggi come "Sono presenti degli errori" oppure "Cruciverba completato con successo".

A questo punto l'aspetto grafico del cruciverba in Flash può dirsi ultimato, ma rimangono da implementare tutti gli script in grado di rendere interattiva l'applicazione e di rendere operative tutte le sue funzionalità. In altre parole è necessario scrivere tutto il codice di scripting all'interno del tag `<interpret>`.

Iniziamo con le caselle del cruciverba: selezionandone una si richiede che vengano stampate le definizioni delle parole che da essa hanno origine nel campo di testo appositamente creato. Per fare ciò è innanzitutto necessario aggiungere un gestore di eventi per ogni casella del cruciverba (quindi il tag `<handlers>` inserito all'interno dei tag `<EditableText>` che rappresentano le caselle), quindi scrivere una funzione che venga azionata nel momento in cui una casella acquisisce il focus, ovvero viene selezionata, ad esempio cliccandoci sopra con il mouse, ed è possibile digitarvi una lettera.

Ecco che quindi viene approntata una funzione per ogni casella, la quale contiene lo script necessario per stampare la o le definizioni relative a tale casella (se si tratta di due, sia orizzontale che verticale, esse vengono opportunamente separate con un trattino) nel campo di testo approntato per tale scopo, aggiungendovi anche informazioni circa il numero e la direzione che le identificano (ad esempio "1 orizzontale: definizione della parola orizzontale – 1 verticale: definizione della parola verticale"). È stato scelto di non applicare tale funzionalità per le caselle prive di definizione, in modo che selezionando una di esse non si verifichi alcun cambiamento nel testo mostrato nel campo di testo delle definizioni. Oltre a ciò la funzione si occupa di colorare in nero il testo della casella, ciò perché, come si vedrà, le lettere possono essere colorate in rosso per segnalare un errore durante lo svolgimento dell'applicazione. In tal modo quando ci si appresta a correggere tale casella, l'eventuale lettera rossa in essa contenuta torna ad essere nera. Questa seconda funzionalità è ovviamente applicata a tutte le caselle del cruciverba, indipendentemente dal fatto che esse siano caratterizzate da definizioni o meno.

È ora la volta dei tre pulsanti aggiunti all'interfaccia grafica dell'applicazione, i quali sono stati infatti per ora solo disegnati e lasciati privi di funzionalità. Le operazioni da svolgere sono in pratica le stesse viste per le caselle del cruciverba: inserire un gestore di eventi all'interno dei tag `<Button>` che rappresentano i tre pulsanti e implementare una funzione con del codice di scripting

per ognuno di essi, la quale viene attivata alla pressione del relativo pulsante, in grado di svolgere le operazioni richieste.

Vediamole in dettaglio: iniziando dal pulsante “Cancella tutto” per la completa cancellazione del cruciverba, la funzione associata alla sua pressione contiene il codice in grado semplicemente di rimuovere il testo da ogni casella `<EditableText>` che compone il cruciverba e di stampare il messaggio “Cruciverba resettato” nel campo di testo approntato per la notifica degli eventi.

La funzione relativa al pulsante “Verifica” per l’analisi dello stato attuale del cruciverba si occupa invece di verificare il contenuto di tutte le caselle `<EditableText>` del cruciverba e confrontarlo con la lettera che dovrebbero contenere per essere corrette. Nel caso in cui una casella contenga una lettera che non corrisponde a quella corretta, essa viene colorata in rosso per mettere in evidenza l’errore. La funzione inizializza inoltre due variabili contenenti un numero intero con valore inizialmente posto a 0, chiamate n ed e . Per ogni casella con il contenuto che non corrisponde a quello previsto, la variabile n viene incrementata di 1, mentre si fa lo stesso per e solo se l’errore è dovuto all’assenza di testo, ovvero se nella casella non è stata ancora digitata alcuna lettera. Quando tutte le caselle sono state analizzate la funzione stampa un messaggio nel campo di testo per la notifica degli eventi; tale messaggio varia a seconda dei valori assunti dalle due variabili numeriche: se n è maggiore di 0 ed anche di e (significa che nel cruciverba sono state digitate delle lettere errate) allora il messaggio sarà “Ci sono degli errori!”, se n è maggiore di 0 ed uguale ad e (ovvero non vi sono lettere errate, ma solo caselle ancora vuote) il messaggio sarà “Fin qui corretto!”, se infine n è uguale a 0 (significa che il cruciverba è stato riempito interamente e senza alcun errore) il messaggio stampato sarà “Completato con successo!”.

Il terzo pulsante, “Vedi soluzione”, è caratterizzato da degli script leggermente più complessi. Il funzionamento che è stato ideato per esso infatti prevede la gestione di due eventi diversi e non solo uno come nei casi precedenti: si vuole che alla pressione del pulsante la soluzione completa del cruciverba venga visualizzata (ovvero vengono stampate tutte le lettere corrette nelle loro relative caselle), mentre nel momento in cui si rilascia la pressione del tasto del mouse, la soluzione appena mostrata deve sparire ripristinando lo stato in cui il cruciverba appariva precedentemente. In pratica la soluzione viene mostrata per il tempo in cui si mantiene la pressione del tasto del mouse sul pulsante. Il gestore degli eventi in questo caso deve quindi prevedere entrambi gli eventi, ognuno con associata la propria funzione: la stampa della soluzione alla pressione del pulsante e la rimozione della soluzione al suo rilascio.

Analizzando in dettaglio l’implementazione delle due funzioni, si nota che la prima (alla pressione del pulsante) svolge tre operazioni: prima di tutto inizializza tante variabili quante sono le caselle che compongono il cruciverba e salva in ognuna di esse il contenuto della relativa casella (in modo

da saper ricostruire lo stato precedente del cruciverba), quindi in tutte le caselle viene inserita la lettera corretta prevista dalla soluzione, sostituendo l'eventuale lettera già digitata dall'utente, ed infine stampa un messaggio nel campo di testo riservato alla notifica degli eventi, ovvero "Soluzione visualizzata".

La seconda funzione (al rilascio del pulsante) invece sostituisce nuovamente il contenuto di tutte le caselle stampandovi ciò che è contenuto nelle variabili salvate dalla precedente funzione, in modo quindi da ripristinare lo stato precedente del cruciverba, e provvede inoltre a rimuovere il messaggio "Soluzione visualizzata" dal campo di testo per la notifica degli eventi.

5.3. Nuove funzionalità

A questo punto dello sviluppo, il documento XML comunica all'applicazione Flash tutte le informazioni necessarie a visualizzare la griglia del cruciverba ed i controlli necessari per il suo utilizzo e ad attivare tutte le funzionalità descritte in precedenza, ottenendo quindi un oggetto Flash interattivo e funzionale.

Fin dal primo test svolto per valutare l'effettiva validità del prodotto finale ci si è resi conto di mancanze dal punto di vista del controllo dell'utente sul cruciverba stesso, infatti una volta selezionata una casella con il mouse e digitata in essa una lettera, risultava necessario selezionare un'altra casella con il mouse per scrivere la lettera successiva, togliendo le mani dalla tastiera e rendendo pertanto estremamente lenta la digitazione anche di una sola parola.

La soluzione ideata consiste nell'implementare la possibilità di spostarsi tra le caselle del cruciverba utilizzando la tastiera ed in particolare i tasti delle frecce direzionali. Oltre a ciò è stato pensato di velocizzare ulteriormente la scrittura di una parola eseguendo uno spostamento nella casella adiacente dopo la digitazione di ogni lettera, in modo da poter scrivere una parola intera senza alcuna interruzione.

L'introduzione di queste nuove funzionalità ha avuto come logica conseguenza la gestione della direzione di scrittura (orizzontale o verticale), in quanto deve essere resa disponibile la possibilità di scegliere il senso in cui digitare le parole.

È stato pensato di offrire all'utente la possibilità di passare da una direzione all'altra tramite la pressione di una combinazione di tasti (in particolare si è optato per il tasto "Control" più una qualsiasi delle quattro frecce direzionali), in modo da permettere di mantenere le mani sulla tastiera senza perdere il tempo necessario all'utilizzo del mouse.

È stato però ritenuto opportuno, per la completezza dell'interfaccia grafica, aggiungere anche un controllo grafico che svolga tale operazione; a tale scopo si è pensato di introdurre un Check Box,

ovvero un elemento grafico (tipicamente un quadratino bianco) cliccando sul quale è possibile attivare e disattivare una funzionalità (tipicamente vi apparirà un segno di spunta in caso di attivazione oppure il quadrato rimarrà vuoto in caso di disattivazione). La disattivazione corrisponderebbe ad esempio alla scrittura orizzontale, l'attivazione a quella verticale. Per rendere inoltre il controllo più immediato, si è pensato di associarvi un nuovo campo di testo che riporta esplicitamente la direzione attualmente attiva. Il testo in esso contenuto chiaramente varia a seconda dell'interazione con il Check Box. Tali elementi sono stati quindi inseriti nell'interfaccia di controllo dell'applicazione, a destra della griglia del cruciverba.

Il Check Box viene identificato dal tag `<SelectBox>` nel codice all'interno del documento XML e, grazie agli attributi tramite i quali è possibile personalizzarlo, si è scelto di farlo apparire come un quadrato bianco con gli angoli elegantemente arrotondati quando deselezionato, mentre il colore diventa blu in caso di selezione. Accanto ad esso è stato piazzato il campo di testo che riporta la direzione visualizzando la parola "Orizzontale" quando il Check Box è deselezionato e "Verticale" quando esso è selezionato.

Rimane ora da scrivere il codice per rendere interattivo tale controllo, ovvero due funzioni associate al Check Box, una che descrive il comportamento dell'applicazione quando quest'ultimo viene selezionato ed una quando esso viene deselezionato. È chiaro che successivamente dovranno essere sviluppati altri script per permettere, come già detto, la scrittura di parole intere senza interruzioni tramite gli spostamenti automatici tra le caselle, e che le funzioni che rendono ciò possibile dovranno avere ad immediata disposizione l'informazione riguardo alla direzione attualmente selezionata; per questo è stata inizializzata nel codice del documento XML una variabile globale (ovvero visibile a tutte le funzioni che compongono il codice di scripting dell'applicazione) che inizialmente conteneva un valore booleano (in particolare "False" se il Check Box è deselezionato e quindi la direzione di scrittura è quella orizzontale e "True" nel caso contrario), poi sostituito da un semplice numero intero (0 in caso di direzione orizzontale, 1 nel caso opposto).

A questo punto è stato inserito nel codice il gestore degli eventi per il Check Box e sono state scritte le due funzioni già citate che fanno riferimento ad esso. Molto semplicemente la funzione associata alla selezione del Check Box si occuperà di stampare nel campo di testo apposito la parola "Verticale" e di settare la variabile globale che indica la direzione ad 1; similmente la funzione attivata alla deselezionazione dell'elemento grafico svolge le operazioni inverse: stampa "Orizzontale" nella casella di testo e setta la variabile globale a 0.

Come permettere lo spostamento tra le caselle del cruciverba alla pressione delle frecce direzionali e alla digitazione di una lettera? E come consentire di cambiare la direzione di scrittura premendo una combinazione di tasti?

Il gestore degli eventi per i componenti `<EditableText>` prevede anche il rilevamento dell'evento rappresentato dalla pressione di un tasto della tastiera, con la possibilità di identificare quale è stato premuto e quindi di attuare strategie diverse a seconda di ciò. Grazie ad esso è possibile implementare le funzionalità richieste: è innanzitutto necessario mettere i gestori di eventi di tutte le caselle del cruciverba in condizione di attivare una funzione anche in risposta alla pressione di un tasto da tastiera. Fatto ciò è stata scritta una nuova funzione per ogni casella con il compito di svolgere alcuni controlli sui tasti premuti e di eseguire alcune operazioni in funzione di essi.

Analizzandone il funzionamento in dettaglio, queste funzioni prima di tutto verificano che si stia tenendo premuto il tasto "Control", nel qual caso passano ad analizzare il tasto che viene premuto successivamente, creando una combinazione. Come è stato detto, quello che si voleva è un cambiamento di direzione di scrittura alla pressione di "Control" più una delle frecce direzionali, ecco quindi che se la funziona rileva proprio tale combinazione va a svolgere stessa funzione del Check Box grafico, ovvero modifica il valore della variabile globale che determina la direzione (passando da 0 ad 1 o viceversa, a seconda dello situazione presente) e cambia la parola contenuta nel campo di testo vicino da "Orizzontale" a "Verticale" o viceversa. Oltre a ciò deve anche cambiare lo stato del Check Box come se vi si fosse cliccato sopra con il mouse, modificandolo da selezionato a deselezionato o viceversa.

Se la funzione ha rilevato la pressione continuata del tasto "Control", qui si conclude il suo compito, infatti in caso di cambiamento di direzione di scrittura non è necessario effettuare anche lo spostamento nella casella adiacente. Al contrario, se non è stata rilevata la pressione del tasto "Control" la funzione analizza il singolo tasto che viene premuto per primo comportandosi di conseguenza: se si tratta di una delle frecce direzionali ci si sposterà nella casella adiacente nella direzione indicata dal tasto (nel caso in cui in quella direzione vi sia una casella annerita e quindi non selezionabile oppure nessuna casella poiché ci si trova sul bordo, non verrà svolta alcuna operazione), assegnandole quindi il focus. È da notare che in tal caso, assegnando il focus ad una nuova casella, verrà automaticamente eseguita la funzione associata a tale evento già descritta in precedenza.

Se il tasto premuto invece non è una delle frecce direzionali, allora, dopo aver stampato nella casella attualmente selezionata il carattere identificato dal tasto premuto, la funzione rileva la direzione di scrittura correntemente attiva basandosi sulla variabile globale appositamente approntata ed in base ad essa esegue uno spostamento nella casella adiacente a destra o sottostante

quella in cui si è digitato il carattere (tale spostamento non viene effettuato nel caso in cui la casella adiacente sia annerita oppure si sia arrivati ad un bordo della griglia del cruciverba).

Infine è da notare che la funzione rileva anche la pressione del tasto “Control” oltre alle frecce, ed in questo caso non opera alcuno spostamento, in modo così da evitare di effettuare uno alla pressione della combinazione di tasti vista in precedenza per il cambiamento di direzione.

5.4. Le ultime modifiche

Giunti a questo punto, ed in luce dei test svolti sull’applicazione Flash risultante, il lavoro poteva dirsi ultimato e tutti i requisiti soddisfatti. Tuttavia ulteriori modifiche e migliorie applicabili sono state individuate e sviluppate, riassumibili nelle seguenti linee guida:

- Reimpaginare l’intera applicazione in modo da avere l’interfaccia di controllo sottostante il cruciverba, mentre a quest’ultimo viene applicato l’allineamento centrato rispetto a quello sinistro;
- Aumentare le dimensioni massime raggiungibili dal cruciverba (precedentemente fissate a 16 per 12 caselle) fino a 20 per 20;
- Migliorare la visualizzazione delle definizioni in modo tale da mostrarne solo una (in relazione alla direzione di scrittura attiva al momento) che venga stampata non solo se si seleziona la casella di origine della parola identificata da tale definizione come avveniva in precedenza, ma bensì una qualsiasi casella che compone la parola;
- Sostituire il controllo per il cambio della direzione di scrittura con un pulsante in modo da rendere l’interfaccia di controllo più uniforme, con ogni funzionalità associata ad un pulsante;
- Visualizzare tutte le lettere nel cruciverba in maiuscolo, così come si è abituati a fare in quelli cartacei e nella maggior parte di quelli già esistenti online.

La modifica dell’impaginazione dell’intera applicazione ha comportato un completo rinnovo del calcolo delle coordinate di tutti i componenti di cui è costituita.

Per quanto riguarda l’allineamento centrato del cruciverba, la coordinata x delle caselle va ora calcolata non più partendo da 0 e poi crescendo, ma bensì partendo da un particolare valore calcolabile nel seguente modo: partendo da un valore pari alla metà della larghezza totale dell’applicazione, è necessario sommare ad esso la metà della larghezza del cruciverba e sottrarvi un valore pari all’intera larghezza del cruciverba meno la larghezza delle caselle già disegnate sulla riga attuale.

Per fare un esempio, la coordinata x della terza casella di una riga si calcolerà in questo modo:

$x = \text{metà della larghezza della schermata dell'applicazione} + \text{metà della larghezza della griglia del cruciverba} - (\text{larghezza della griglia del cruciverba} - \text{somma della larghezza delle prime due caselle della riga})$

In tal modo il cruciverba verrà disegnato esattamente al centro della schermata visualizzata dall'applicazione.

L'interfaccia di controllo invece, essendo riposizionata in posizione sottostante la griglia del cruciverba, deve vedere la sua coordinata y rielaborata in funzione dell'altezza del cruciverba stesso, in modo da apparire immediatamente sotto di esso indipendentemente da quante caselle lo costituiscano.

Anche l'aumento delle dimensioni massime previste per il cruciverba hanno come conseguenza una problematica: se le dimensioni di una singola casella rimanessero invariate, l'intera griglia sarebbe più grande della schermata a disposizione dell'applicazione Flash sui calcolatori con settata una risoluzione media o bassa.

La soluzione più semplice per questo problema è quella di ridurre le dimensioni delle singole caselle in modo da assicurarsi che anche un cruciverba 20 per 20 sia contenuto in una schermata, ma così facendo quelli più piccoli (ad esempio 5 per 5) risulterebbero esageratamente piccoli rispetto all'intera interfaccia. È stata quindi scelta un'altra soluzione, ovvero quella di ridurre le dimensioni delle caselle solo nei cruciverba più grandi.

È chiaro che il compito di svolgere tutti i calcoli riguardanti dimensioni e coordinate dei componenti grafici del cruciverba Flash è dell'applicazione Java che dovrà generare il file XML in cui tutti i dati vengono inseriti; queste problematiche verranno quindi riprese successivamente.

Per quanto riguarda la visualizzazione delle definizioni relative alla casella selezionata, come già detto si è deciso di stamparne solo una, ovvero solo quella relativa alla parola che si sta digitando (a seconda della direzione in cui si scrive), indipendentemente in quale casella che compone la parola ci si trovi.

Ciò comporta alcuni cambiamenti da applicare al codice di scripting: innanzitutto tutte le caselle vengono ad avere almeno una definizione associata (ad esempio se abbiamo una parola di cinque lettere orizzontale, tutte le cinque caselle che la compongono presenteranno la stessa definizione orizzontale e non soltanto la prima, ovvero quella in cui la parola ha origine), quindi le funzioni che vengono eseguite quando una casella acquisisce il focus non stampano più tutte le definizioni che

caratterizzano tale casella (siano esse una o due), ma effettuano prima un controllo sulla direzione di scrittura attiva al momento, stampando quindi la definizione relativa a tale orientamento.

Possono ancora esistere tuttavia caselle che presentano una sola definizione, poiché non contengono una lettera facente parte di una parola scritta nel senso opposto; basti pensare ad una parola orizzontale che si trova nell'ultima riga del cruciverba e che si conclude nella casella che sta nell'angolo in basso a destra. Se la casella sovrastante quella d'angolo è annerita, l'unica definizione associata all'angolo è quella orizzontale. In questo caso si è deciso di visualizzare l'unica definizione disponibile indipendentemente dalla direzione di scrittura attiva.

Un'altra considerazione da fare riguarda la funzionalità che permette il cambio di direzione di scrittura: in caso di cambiamento di orientamento infatti, automaticamente anche la definizione visualizzata deve cambiare.

Per ottenere ciò è necessario applicare delle modifiche nelle funzioni in cui si gestisce la direzione di scrittura, ovvero quelle associate al controllo Check Box approntato proprio per tale scopo e quelle eseguite ogni volta che viene premuto un tasto mentre una casella è selezionata.

Per quanto riguarda il cambio di orientamento tramite la combinazione di tasti "Control" più una freccia direzionale, gestito nelle funzioni associate alla pressione di un tasto mentre ci si trova in una delle caselle del cruciverba, è sufficiente permettere ad esse, dopo aver completato tutte le operazioni per l'effettivo cambiamento di direzione, di richiamare ed eseguire la funzione associata all'acquisizione del focus da parte della casella in cui si sta operando. In pratica si simula una nuova selezione della stessa casella in cui già ci si trova e di conseguenza la funzione che gestisce l'acquisizione del focus stamperà la definizione relativa alla direzione di scrittura appena modificata.

Nel caso del Check Box la soluzione non è altrettanto semplice poiché dal momento in cui si va ad interagire con esso tramite il mouse non vi sono più caselle selezionate e non si conoscono informazioni circa quella selezionata in precedenza. Per questo se la direzione di scrittura viene cambiata non è possibile richiedere l'esecuzione della funzione per l'acquisizione del focus di una casella, poiché non vi è una casella specifica su cui svolgere tale operazione.

La soluzione ideata è la seguente: vengono inizializzate due variabili globali contenenti il testo delle definizioni attive al momento, rispettivamente orizzontale e verticale. All'avvio dell'applicazione esse sono vuote, ma non appena una casella acquisisce il focus, la funzione associata a tale evento vi inserisce le due definizioni relative a tale casella, e così ogni volta che una nuova casella viene selezionata. Nel momento in cui si interagisce con il Check Box, nonostante non si abbiano più informazioni riguardanti le caselle, ci si può basare su tali variabili per cambiare la definizione visualizzata al cambiamento di direzione di scrittura. In altre parole è come se l'ultima casella

selezionata prima dell'interazione con il Check Box sia ancora attiva e si abbia accesso alle sue due definizioni.

Si è scelto poi di sostituire l'intero controllo Check Box con un pulsante riportante la dicitura "Cambia" principalmente per avere, come detto, una maggiore uniformità nell'interfaccia di controllo. Questa modifica ha inoltre permesso la semplificazione del codice di scripting: innanzitutto le due funzioni in precedenza eseguite rispettivamente alla selezione ed alla deselegione del Check Box vengono fuse in una singola, attivata ovviamente alla pressione del nuovo pulsante; essa rileva l'attuale orientamento di scrittura (basandosi sull'apposita variabile globale) ed in base ad essa svolge esattamente le funzionalità di una delle due vecchie funzioni, ovvero la modifica della variabile globale che indica la direzione attiva ed il contenuto del campo di testo che mostra a seconda del caso le parole "Orizzontale" o "Verticale".

Inoltre va applicata una modifica anche nelle funzioni eseguite alla pressione di un tasto mentre una casella è selezionata: nel caso in cui venga premuta la combinazione di tasti "Control" più una freccia direzionale per cambiare la direzione di scrittura in precedenza si dovevano eseguire le stesse operazioni svolte dalle funzioni di selezione e deselegione del Check Box oltre alla modifica dello stato di tale componente grafico (da selezionato a non e viceversa). Tutto ciò ora viene rimpiazzato da un semplice richiamo della stessa funzione eseguita alla pressione del nuovo pulsante, infatti non serve più alcuna operazione ulteriore rispetto a quelle da essa svolte.

Infine rimane da applicare la trasformazione di tutte le lettere digitate nel cruciverba da minuscole (la modalità di scrittura predefinita) a maiuscole. È chiaramente necessario far eseguire questa operazione all'applicazione ogniqualevolta una nuova lettera viene inserita in una casella, per questo le funzioni a cui è stato assegnato tale compito sono quelle eseguite alla pressione di un tasto mentre una casella è selezionata.

Prima di eseguire ogni altra operazione a loro già assegnata, ora esse, non appena un tasto viene premuto, si occupano di trasformare il testo contenuto nella casella ad esse associata, facendolo diventare maiuscolo. C'è da dire che le funzioni associate al pulsante per la visualizzazione della soluzione del cruciverba si occupano, come visto, di stampare tutte le lettere corrette nel cruciverba per poi rimuoverle e ripristinare quelle precedentemente presenti, ma in questo caso non è necessario richiedere la trasformazione del testo poiché tali funzioni stampano le lettere già in forma maiuscola, senza quindi la necessità di farlo in un secondo momento a stampa avvenuta.

Con ciò si è giunti alla versione finale del cruciverba in Flash (Fig.2).

1	I	2	N	3	D	4		5
6	C			7			8	
	E		9			10		
		11						12
13								
14								
		15						16
			17					
18						19		

Definizione attiva:

2 verticale: Numero in breve

Direzione della scrittura:

VERTICALE

Cambia

Controlli:

Verifica

Cancela tutto

Vedi soluzione

Fig.2: Screenshot del cruciverba in Flash

5.5. La struttura del file XML

Il file XML si può definire concluso; successivamente sarà l'applicazione Java per la creazione dei cruciverba a dover generarlo automaticamente sulla base di questo primo scritto a mano. È bene riassumerne la struttura finale per punti in modo da facilitarne la lettura:

1. `<Content id="master">`

Inizialmente si trova il componente radice dell'intero file, quello che contiene tutti gli altri, identificato dal tag `<Content>` e denominato *master*.

2. `<style>`
`EditableText {`
`font-family: Arial,Helvetica;`
`font-size: 16;`
`text-align: center;`
`}`
`</style>`

In seguito vi è la parte relativa alla formattazione delle lettere che vengono digitate nelle caselle del cruciverba, la quale sfrutta la tecnologia CSS supportata da Gaupe. È contenuta all'interno del tag `<style>`.

3. `<Text id="con" x="360" y="z+65" fontSize="14" fontWeight="bold"> Controlli:</Text>`

```

<Text id="msg" x="750" y="z+80" color="#0000FF" fontSize="16"></Text>
<Text id="def_act" x="20" y="z+15" fontSize="14" fontWeight="bold"> Definizione attiva:</Text>
<Text id="act" x="20" y="z+35" color="#0000FF" fontSize="16">Selezionare una casella</Text>
<Text id="tog" x="20" y="z+65" fontSize="14" fontWeight="bold">Direzione della scrittura:</Text>
<Text id="d" x="20" y="z+90">ORIZZONTALE</Text>

```

La terza parte contiene i dati necessari per disegnare tutte le caselle di testo presenti nell'interfaccia di controllo, ognuna rappresentata da un tag `<Text>`. La prima, denominata *con* e non citata in precedenza, si limita ad introdurre tutti i controlli presenti nell'interfaccia; la seconda, *msg*, è quella con il compito di riportare le notifiche degli eventi ed inizialmente non contiene alcun testo; la terza, *def_act*, introduce la definizione attiva visualizzata subito sotto ad essa; la quarta, *act*, è quella in cui verrà mostrata la definizione attiva durante lo svolgimento del cruciverba, ed all'avvio dell'applicazione vi si legge semplicemente "Selezionare una casella"; la quinta, *tog*, introduce la direzione di scrittura attiva, riportata sotto ad essa; la sesta infine, *d*, riporta la l'orientamento con cui si scrive al momento (il valore predefinito è "Orizzontale" come si può notare). Tutti i valori riportati qui sopra sono costanti ad eccezione del numero *z*, un numero al quale viene sommato un valore costante per trovare la coordinata *y* dei campi di testo. Tale valore *z* è in realtà un numero intero variabile a seconda delle dimensioni del cruciverba; sarà compito dell'applicazione Java calcolarlo per ogni cruciverba generato.

```

4. <Button id="sub" padding="2" size="12" fontWeight="bold" fontAlign="center"
background-color="#F6F6F6" overColor="#000000" color="#666666" x="280" y="z+90" width="100"
height="20" value="Verifica" roundness="4" border="1" borderColor="#333333" borderAlpha="100">
<handlers release="onReleaseSubmit"/>
</Button>
<Button id="res" padding="2" size="12" ... x="410" y="z+90" width="100" height="20" value="Cancella
tutto" ...>
<handlers release="onReleaseReset"/>
</Button>
<Button id="sol" padding="2" size="14" ... x="570" y="z+75" width="130" height="30" value="Vedi
soluzione" ...>
<handlers press="onPressSolution" release="onReleaseSolution"/>
</Button>
<Button id="dir" padding="2" size="12" ... x="130" y="z+90" width="80" height="20" value="Cambia" ...>
<handlers release="onReleaseDirection"/>
</Button>

```

La quarta sezione contiene i quattro pulsanti presenti nell'interfaccia di controllo dell'applicazione, ognuno dei quali è rappresentato da un'istanza del tag `<Button>`. Da notare in ognuno di essi il tag `<handlers>`, ovvero il gestore degli eventi, con la o le funzioni ad esso associate. Il primo pulsante, denominato *sub*, è quello per azionare l'analisi dello stato del cruciverba, ottenendo la notifica degli errori ed un messaggio esplicativo, operazioni svolte dalla funzione *onReleaseSubmit*; il secondo, *res*, serve per resettare il cruciverba, ovvero cancellarvi tutte le lettere digitate, grazie alla funzione *onReleaseReset*; il

terzo, *sol*, è il pulsante per la visualizzazione e la successiva rimozione della soluzione del cruciverba tramite le due funzioni che svolgono tali compiti, ovvero *onPressSolution* e *onReleaseSolution*; il quarto e ultimo, *dir*, è quello per cambiare la direzione di scrittura attiva tramite la funzione *onReleaseDirection*. Ogni pulsante è descritto da molti attributi costanti che ne stabiliscono i dettagli grafici come dimensioni, colori e forma (sono stati riportati al completo soltanto per il primo pulsante). Come per i campi di testo l'unico valore variabile è nuovamente *z* che dipende dalle dimensioni del cruciverba.

```
5. <Drawing id="black">
  <quad x="3" y="2" w="28" h="27" color="#000000" alpha="100"/>
  <line x="1" y="-1" w="32" h="0" color="#000000" alpha="100"/>
  <line x="1" y="31" w="32" h="0" color="#000000" alpha="100"/>
  <line x="0" y="-1" w="0" h="33" color="#000000" alpha="100"/>
  <line x="33" y="-1" w="0" h="33" color="#000000" alpha="100"/>
</Drawing>
```

La parte successiva è quella che costruisce il simbolo Flash le cui istanze rappresenteranno le caselle annerite, introdotto dal tag `<Drawing>`. Come si può notare tale oggetto è denominato *black* ed è composto da cinque elementi grafici: un quadrilatero identificato dal tag `<quad>` che raffigura il quadrato nero all'interno della casella annerita e quattro linee rappresentate dal tag `<line>` che realizzano i bordi della casella annerita. Da notare che le coordinate di questo componente grafico sono relative, in quanto esso verrà disegnato a tutti gli effetti solo quando richiesto all'interno di altri componenti (i padri). Queste coordinate dovranno quindi essere sommate a quelle del componente padre per avere il risultato finale.

```
6. <Text id="a1_1" x="_global.Stage.width/2+g-n" y="0" width="2" height="18" fontSize="10">1</Text>
  <EditableText id="c1_1" x="_global.Stage.width/2+g-n" y="1" width="w" height="h" maxLength="1"
  multiLine="true" wordWrap="true" border="1" borderColor="#000000" borderAlpha="100">
  <handlers focus="onSelectCell1_1" keyup="onKeyUp1_1"/>
  </EditableText>
  <Text id="c3_2" x="_global.Stage.width/2+g-m" y="65" width="w" height="h">
  <draw id="black"/>
</Text>
```

La successiva parte del codice XML contiene i dati per disegnare la griglia del cruciverba casella per casella. Qui sopra è riportato il testo che descrive due caselle, la prima normale e la seconda annerita. Come detto in precedenza una casella normale è composta da un piccolo campo di testo (tag `<Text>`) contenente il numero che identifica le definizioni associate alla casella stessa (in questo caso il numero è 1) e da un campo di testo di input (tag `<EditableText>`) dove vi si digita la lettera dotato di bordo nero e di un gestore degli eventi (tag `<handlers>`) con due funzioni, *onSelectCell* per l'acquisizione del focus e *onKeyUp* per la pressione di un tasto. Tra gli attributi che descrivono il campo di testo di input, oltre a quelli grafici vi è anche quello che permette la digitazione di un solo carattere nella casella,

ovvero *maxLength*. Una casella annerita invece è composta da un campo di testo vuoto il quale fa da padre ad un'istanza del simbolo che raffigura la casella annerita (disegnata tramite il tag `<draw>`). Da notare gli indici numerici presenti negli identificatori dei campi di testo e nel nome delle funzioni (negli esempi qui sopra `1_1` e `3_2`); essi indicano riga e colonna di appartenenza della casella. In questo esempio quindi sono state prese in esame la casella nell'angolo in alto a sinistra e la seconda della terza riga. Per quanto riguarda le coordinate dei componenti, si tratta di valori variabili a seconda delle dimensioni del cruciverba e della posizione delle caselle: il valore di *y* sarà pari a 0 per le caselle della prima riga e aumenterà per quelle alle righe successive a seconda dell'altezza delle caselle stesse, mentre *x* si calcola trovando la metà della larghezza della schermata dell'applicazione, sommandovi la metà della larghezza della griglia del cruciverba (la variabile *g* nell'esempio sopra) e sottraendovi un valore pari alla larghezza totale della griglia del cruciverba meno quella delle caselle già inserite nella riga attuale (le variabili *n* ed *m* nell'esempio). Anche gli attributi che indicano altezza e larghezza delle caselle sono variabili a seconda delle dimensioni del cruciverba (le variabili *w* e *h* nell'esempio).

7. `<interpret>`

Gli elementi grafici sono terminati; da questo punto in poi nel file XML si trova esclusivamente del codice che verrà interpretato da Gaupe per applicare l'interattività all'applicazione Flash risultante. Il tag `<interpret>` demarca l'inizio di tale codice.

```
8. var _global.ov = 0;
   var _global.defori = "vuota";
   var _global.defver = "vuota";
```

Il codice per l'interattività dell'applicazione Flash è strutturato in funzioni le quali vengono eseguite al presentarsi di un determinato evento come la pressione di un pulsante, ma prima di esse vengono inizializzate tre variabili globali che possono essere utilizzate, cioè lette e modificate, da tutte le funzioni presenti. La prima di esse, *ov*, indica la direzione di scrittura attiva al momento. Il valore 0 (quello predefinito) sta ad indicare scrittura in senso orizzontale, 1 invece sta per verticale. Le altre due variabili contengono rispettivamente le definizioni orizzontale e verticale dell'ultima casella selezionata, in modo tale da non "dimenticarle" nel momento in cui la selezione viene rimossa. All'avvio dell'applicazione, come si può notare, esse non contengono dati significativi fino a che la prima casella non viene selezionata.

```
9. function onReleaseDirection() {
   if ( _global.ov == 0 ) {
```

```

    _global.ov = 1;
    d.setText("VERTICALE");
    act.setText( _global.defver );
}
else if ( _global.ov == 1 ) {
    _global.ov = 0;
    d.setText("ORIZZONTALE");
    act.setText( _global.defori );
}
}
}

```

La prima funzione che si incontra è quella eseguita in seguito alla pressione del pulsante per il cambiamento della direzione di scrittura. Per prima cosa essa verifica l'orientamento attualmente attivo ed a seconda di ciò lo cambia da orizzontale a verticale o viceversa. In pratica essa modifica la variabile globale che indica la direzione attiva, la parola mostrata nel campo di testo *d* che riporta tale direzione ed il contenuto del campo di testo in cui viene visualizzata la definizione attiva, stampandovi quella corretta tra le due contenute nelle variabili globali viste in precedenza.

```

10. function onReleaseReset() {
    c1_1.setText("");
    ...
    msg.setText("Cruciverba resettato");
}

```

Segue la funzione per la cancellazione del cruciverba. Le operazioni da essa svolte sono la rimozione del testo contenuto in ogni casella del cruciverba (nell'esempio ciò è riportato solo per la prima) e la stampa di un messaggio nel campo di testo per le notifiche.

```

11. function onReleaseSubmit() {
    var n = 0;
    var e = 0;
    if (c1_1.getText()!="P") {
        c1_1.tf.color = "0xFF0000";
        c1_1.field.setTextFormat(c1_1.tf);
        n = n + 1;
        if (c1_1.getText()=="") {
            e = e + 1;
        }
    }
    ...
    if (n > 0) {
        if (n == e) {
            msg.setText("Fin qui corretto!");
        }
        else {
            msg.setText("Ci sono degli errori!");
        }
    }
    else {
        msg.setText("COMPLETATO CON SUCCESSO!");
    }
}
}

```

La successiva è la funzione per l'analisi dello stato del cruciverba, ovvero per la verifica di eventuali errori commessi durante la sua risoluzione. Analizzando il codice che la compone si può notare come per prima cosa vengano inizializzate due variabili numeriche n ed e , segue quindi un controllo per ogni casella che compone il cruciverba (qui sopra riportato solo quello relativo alla prima) in cui si verifica che esse contengano la lettera prevista dalla soluzione corretta (nell'esempio riportato si tratta della lettera P). Le operazioni svolte nel caso in cui ci si trovi davanti ad un errore sono la colorazione in rosso del testo presente nella casella che si sta analizzando, l'incremento di un'unità della variabile n ed anche di e nel caso in cui l'errore sia dovuto all'assenza di lettere digitate nella casella. Finiti tali controlli la funzione stampa un messaggio nel campo di testo di notifica a seconda dei valori assunti dalle due variabili numeriche: nel caso in cui n sia maggiore di 0, se anche e ha il suo stesso valore allora il cruciverba è incompleto ma privo di errori, altrimenti significa che sono stati commessi errori; se invece n è pari a 0 il cruciverba è stato completato senza alcun errore.

```
12. function onPressSolution() {
    var _global.l1_1 = c1_1.getText();
    ...
    c1_1.setText("P");
    ...
    msg.setText("Soluzione visualizzata");
}
```

È ora la volta delle due funzioni associate al pulsante per la visualizzazione della soluzione del cruciverba; la prima è quella che esegue effettivamente la stampa di tale soluzione. Come si può notare nel codice, per ogni casella le operazioni svolte sono due (di nuovo nell'esempio è riportato solo il caso della prima casella): per prima cosa viene inizializzata una variabile globale e vi viene memorizzato il contenuto della casella presa in esame, quindi in quest'ultima viene stampata la lettera corretta (nell'esempio si tratta di P) sostituendo qualsiasi carattere in essa precedentemente contenuto. Infine viene stampato un messaggio di notifica nell'apposito campo di testo.

```
13. function onReleaseSolution() {
    c1_1.setText(_global.l1_1);
    ...
    msg.setText("");
}
```

La seconda funzione associata al pulsante per la stampa della soluzione è quella che si occupa di rimuovere la stessa ripristinando lo stato precedente del cruciverba. Ciò è ottenuto molto semplicemente stampando in ogni casella il contenuto della variabile globale ad essa

associata approntata dalla funzione precedente (l'esempio riporta il caso della prima casella). Infine viene rimosso il messaggio "Soluzione visualizzata" stampato nel campo di testo di notifica.

```
14. function onSelectCell1_1() {
    msg.setText("");
    c1_1.tf.color = "0x000000";
    c1_1.field.setTextFormat(c1_1.tf);
    if ( _global.ov == 0 ) {
        act.setText("1 orizzontale: ...");
    }
    else if ( _global.ov == 1 ) {
        act.setText("1 verticale: ...");
    }
    _global.defori = "1 orizzontale: ...";
    _global.defver = "1 verticale: ...";
}
```

A questo punto del file XML vi è l'elenco delle funzioni, una per ogni casella del cruciverba, incaricate di svolgere tutte le operazioni necessarie quando una casella viene selezionata, ovvero quando essa acquisisce il focus. Qui viene riportata per esempio quella relativa alla prima casella. Innanzitutto viene rimosso l'eventuale testo contenuto nel campo di testo di notifica (utile nel caso in cui prima della selezione della casella sia avvenuto un evento in seguito al quale un messaggio è stato stampato), quindi la colorazione del testo della casella viene reimpostata di tonalità nera (nel caso in cui essa fosse divenuta rossa in seguito alla rilevazione di un errore). Quindi la funzione si occupa di stampare nell'apposito campo di testo la definizione relativa alla casella in cui si sta operando, scegliendo tra quella orizzontale e quella verticale in base alla direzione di scrittura attiva al momento, determinata dall'analisi della variabile globale che la rappresenta. L'ultima operazione svolta da queste funzioni è infine quella di inserire le due definizioni che caratterizzano la casella nelle due apposite variabili globali, in modo da renderle disponibili alle funzionalità che necessitano di tale informazioni.

```
15. function onKeyUp1_1( id , code ) {
    c1_1.field.text = c1_1.field.text.toUpperCase();
    if ( _global.Key.isDown( _global.Key.CONTROL ) ) {
        if ( (code == _global.Key.RIGHT) || (code == _global.Key.LEFT) || (code == _global.Key.UP) || (code == _global.Key.DOWN) ) {
            onReleaseDirection();
            onSelectCell1_1();
        }
    }
    else {
        if (code == _global.Key.RIGHT) _global.Selection.setFocus(c1_2.field);
        else if (code == _global.Key.DOWN) _global.Selection.setFocus(c2_1.field);
        else if (code == _global.Key.LEFT) {}
        else if (code == _global.Key.UP) {}
        else if (code == _global.Key.CONTROL) {}
    }
}
```

```

else if (_global.ov == 0) _global.Selection.setFocus(c1_2.field);
else if (_global.ov == 1) _global.Selection.setFocus(c2_1.field);
}
}

```

Dopo le funzioni per l'acquisizione del focus, vi sono quelle che vengono eseguite alla pressione di un tasto, di nuovo una per ogni casella del cruciverba. Come in precedenza viene riportato l'esempio per la prima casella. La prima operazione che viene eseguita è la trasformazione in maiuscolo del testo contenuto nella casella. Successivamente viene verificata la pressione del tasto "Control" e di quella contemporanea di una freccia direzionale: in quel caso si richiama prima la funzione che si occupa di cambiare la direzione di scrittura, quindi quella di acquisizione della casella in cui si sta operando. In tal modo il risultato è l'inversione della direzione della scrittura e la stampa della definizione corretta rispetto al nuovo orientamento. Se invece il tasto "Control" non è mantenuto premuto, si verifica quale altro tasto viene battuto: nel caso si tratti di una freccia direzionale si sposta il focus nella casella adiacente a quella attuale seguendo la direzione indicata dalla freccia premuta stessa (a meno che non si cerchi di spostarsi in una casella annerita o attraverso i bordi del cruciverba; nell'esempio, trattandosi della casella nell'angolo in alto a sinistra, come si può notare ci si può spostare solo verso destra o verso il basso); nel caso del tasto "Control" stesso si specifica di non svolgere alcuna operazione; se si tratta di qualsiasi altro tasto, dopo la stampa del carattere relativo nella casella, si effettua uno spostamento nella casella adiacente a seconda dell'orientamento di scrittura attivo (a destra in caso di orientamento orizzontale o in basso in caso di orientamento verticale).

16. </interpret>

Qui il codice riguardante l'interattività dell'applicazione ha termine, perciò viene chiuso il tag <interpret> in cui esso è contenuto.

17. </Content>

L'intero codice che compone il file XML termina qui, per cui come ultima riga viene chiuso anche il componente radice dell'intero documento, ovvero l'elemento denominato *master* visto nel primo punto di questo elenco.

Così si conclude l'analisi del file XML che l'applicazione Java per la generazione di cruciverba deve produrre e che Gaupe deve leggere ed interpretare per ottenere il cruciverba in Flash finale.

5.6. L'applicazione Java

Lo sviluppo dell'applicazione Java ha avuto inizio solo dopo la prima stesura del file XML che avrebbe dovuto produrre come output finale, in modo da conoscere le informazioni che avrebbe dovuto raccogliere per avere a disposizione tutti i dati necessari per la generazione del file XML stesso. Da allora le lavorazioni di una e dell'altro hanno proceduto in parallelo, le modifiche di uno riflesse sull'altra.

Per prima cosa è bene riportare i requisiti che sono stati individuati per l'applicazione; essa deve:

- Raccogliere tutti i dati necessari per la costruzione del file XML che poi sarà utilizzato da Gaupe per la realizzazione del cruciverba in Flash;
- Presentare un'interfaccia grafica chiara ed immediata tramite la quale l'utente non abbia difficoltà ad inserire tutti i dati necessari;
- Consentire un'elevata possibilità di personalizzazione in tutti gli aspetti del cruciverba;
- Facilitare al massimo la realizzazione del cruciverba tramite un sistema di controllo che risulti rapido e funzionale.

Prima di avviare l'implementazione di questa applicazione è stata ideata l'architettura che avrebbe dovuto caratterizzarla.

Da questo punto di vista è stata scelta un design multistrato (in inglese *multi-tier* o *n-tier*) in modo tale da suddividere in più moduli l'intera mole di codice di cui essa sarebbe stata composta, ognuno dei quali in grado di essere sviluppato e mantenuto indipendentemente dagli altri. In particolare si è pensato di applicare un'architettura a tre strati (*3-tier* o *three-tier*) (Fig.3), probabilmente la più popolare, la quale prevede la presenza di un livello di presentazione (*presentation tier*), di uno logico (*logic tier* o *business tier*) e di uno per i dati (*data tier*), ognuno dei quali può comunicare solo con quello o quelli adiacenti all'interno della pila che rappresenta l'architettura:

- **Presentation tier**: è il livello che sta più in alto nella pila che raffigura l'architettura, ovvero quello con cui l'utente viene a contatto e con il quale interagisce. È composto dall'interfaccia grafica ed il suo compito è quello di tradurre i dati e le operazioni svolte dall'applicazione in informazioni leggibili dall'utente.
- **Logic/Business tier**: è lo strato intermedio nella pila dell'architettura e rappresenta il motore vero e proprio dell'applicazione. I suoi compiti sono quelli di processare comandi, svolgere calcoli, valutazioni e controlli e trasferire dati tra i due livelli che stanno sopra e sotto ad esso.

- **Data tier:** è il livello che sta alla base della pila dell'architettura e rappresenta il magazzino dei dati gestiti dall'applicazione. Si tratta in pratica del database dal quale il livello logico sovrastante può leggere ed inserire i dati necessari.

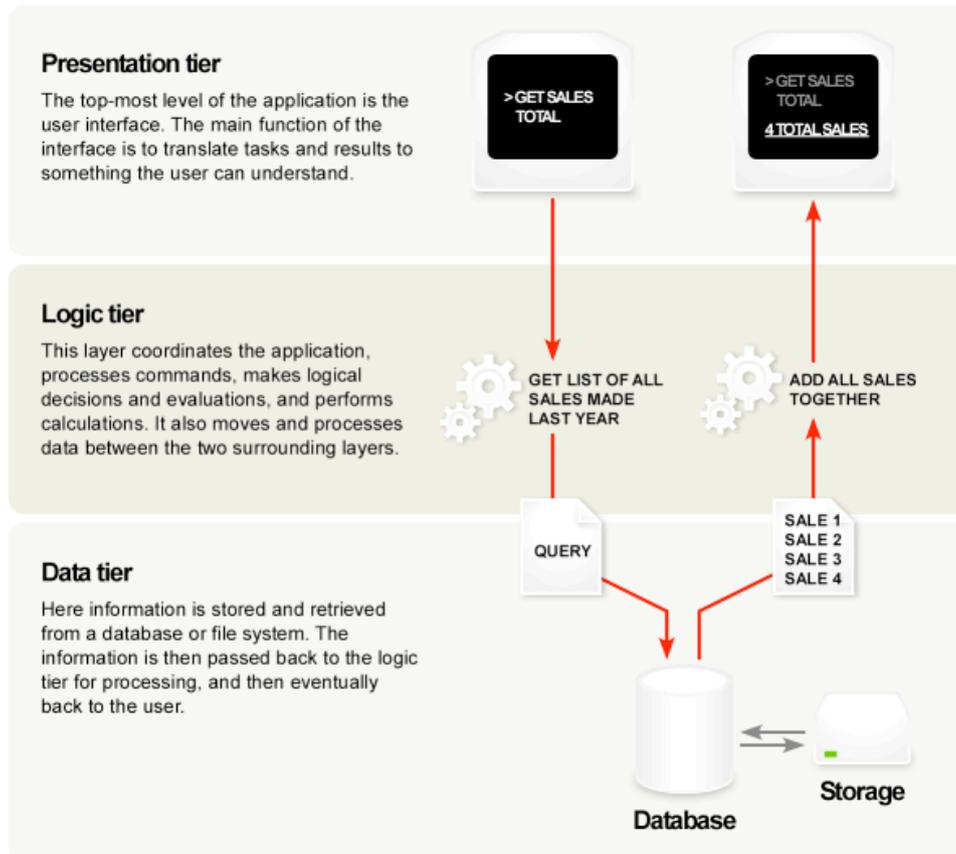


Fig.3: Modello di un'applicazione con architettura *three-tier*

La struttura di base dell'applicazione è stata quindi schematizzata con un database contenente tutti i dati riguardanti ogni casella che compone il cruciverba, un'interfaccia grafica chiara e funzionale che permetta di inserire tutti i dati necessari in maniera semplice ed intuitiva ed infine una o più classi Java che gestiscano tali dati, vi svolgano tutte le elaborazioni necessarie e creino il documento XML basandosi su di essi.

Il primo modulo che si è andati ad implementare è quello dell'interfaccia grafica, per lo sviluppo della quale sono stati elencati gli elementi ed i controlli che essa avrebbe dovuto mettere a disposizione. In un primo momento sono stati individuati i seguenti:

- Una rappresentazione del cruciverba contenente un numero di caselle pari alle sue dimensioni massime (inizialmente 16 per 12) costituite da campi di testo in cui fosse

- possibile digitare la lettera corrispondente, con la possibilità di operare su di esse per annerirle;
- Un controllo che permetta all'utente di scegliere le dimensioni da applicare al cruciverba che si sta costruendo e che in base a tale scelta per esempio disabiliti le caselle in eccesso nella rappresentazione grafica;
 - Un controllo che consenta di indicare quali caselle vanno marcate con un numero, ovvero quali contengono la prima lettera che compone una parola;
 - La possibilità di inserire la o le due definizioni relative alle caselle marcate come origine di una o due parole;
 - La possibilità di assegnare un nome al cruciverba generato, in modo da salvarlo in maniera tale da essere riconoscibile tramite tale nome;
 - Un controllo che avvii la generazione del file XML e quindi dell'applicazione Flash a modellazione del cruciverba ultimata;
 - Voci di menù contenenti informazioni sull'applicazione e istruzioni per il suo utilizzo.

Ha avuto quindi inizio la modellazione dell'interfaccia grafica, in modo tale che essa racchiudesse in sé tutte le funzionalità sopra elencate, utilizzando le API Swing di Java ed operando in ambiente JBuilder di Borland, un IDE per lo sviluppo di Java. Tutte le informazioni su di essa vengono immagazzinate in una classe Java archiviata nel package "presentation", ovvero quello corrispondente al *presentation tier* all'interno dell'architettura *three-tier* adottata.

Si è optato per un'impaginazione simile a quella che caratterizzava la prima versione del cruciverba stesso in Flash, ovvero la griglia delle caselle sulla sinistra della schermata affiancata sulla destra da tutti i controlli esterni. Al di sopra di questa struttura si è pensato di posizionare un'etichetta di benvenuto nel programma, mentre in posizione sottostante è stato riservato lo spazio per l'inserimento del nome del cruciverba ed i principali controlli per l'avvio della generazione dell'applicazione Flash e per l'uscita dal programma. Infine è stato previsto l'inserimento di un menù contenente alcune voci tramite le quali attivare altrettante funzionalità.

Osservando in dettaglio lo struttura dell'interfaccia grafica, iniziamo subito col notare i quattro pannelli principali in cui essa è suddivisa: i due principali, uno per la griglia del cruciverba ed uno per i controlli a destra di esso, rispettivamente con orientamento ovest ed est; quello contenente il messaggio di benvenuto sopra ad essi, quindi con orientamento nord; quello per i controlli principali dell'applicazione che sta sotto ad essi, con orientamento sud.

- **Pannello ovest (griglia):** In questo pannello, il principale dell'intera interfaccia, vengono inseriti 192 campi di testo, ognuno dei quali rappresenta una casella del cruciverba, organizzati a griglia di dimensioni 16 x 12 (quelle massime previste per il cruciverba). Per ognuna di esse vengono settate le dimensioni in modo tale da dare loro forma quadrata e da ottenere una griglia di dimensioni tali da apparire gradevole ed elegante nel complesso. Viene inoltre applicata una formattazione uniforme per il testo in esse contenuto (nonostante all'inizio siano tutte ovviamente vuote), caratterizzato da dimensioni sufficientemente grandi per un'immediata visualizzazione e da un allineamento centrato rispetto ad entrambe le dimensioni del campo di testo, in modo che una lettera digitata appaia esattamente nel centro della casella. Nel momento dell'avvio dell'applicazione tutte le caselle possiedono le stesse proprietà: sono prive di testo ma è possibile digitarvene (sono cioè abilitate, l'utente può interagirvi) e non sono annerite (dovrà essere possibile fare ciò, ma solo dopo l'avvio dell'applicazione);
- **Pannello est (controlli del cruciverba):** Esso viene riservato alla realizzazione di due importanti funzionalità, ovvero la scelta delle dimensioni del cruciverba che si vuol creare ed il settaggio di una casella come origine di una o due parole, con conseguente inserimento della o delle definizioni. Ad ognuna di queste due funzionalità viene a loro volta assegnato un pannello contenuto in quello principale, organizzati in modo tale da apparire uno sopra l'altro (con orientamento rispettivamente nord e sud) nell'ambito del pannello loro padre. In posizione più elevata viene posizionato il controllo per la scelta delle dimensioni del cruciverba; esso è costruito principalmente da due componenti grafici Combo Box, ovvero liste di elementi dalle quali è possibile selezionarne uno per attivarlo, ognuno dei quali rappresenta una delle due dimensioni, larghezza ed altezza. Gli elementi selezionabili in tali liste sono chiaramente i numeri che rappresentano la quantità di caselle ed i valori in essi contenuti vanno da 3 a 16 per quanto riguarda la larghezza, mentre da 3 a 12 per l'altezza. I valori selezionati di predefinito all'avvio dell'applicazione sono quelli massimi in quanto tutte le caselle della griglia sono attive; ciò che si vuole ottenere, come si vedrà in seguito, è la disabilitazione delle caselle in eccesso come conseguenza della selezione di valori inferiori a quelli di partenza ed eventualmente la loro riabilitazione in caso di ritorno a valori maggiori. Nel pannello in esame sono inoltre posizionate alcune etichette testuali descrittive che accompagnano i due Combo Box, una con lo scopo di introdurre l'intero controllo ed altre due che indicano a cosa si riferiscono i valori numerici contenuti nei due Combo Box, ovvero "Larghezza" ed "Altezza". Il secondo pannello, sottostante quello appena visto, ha la principale funzione di permettere la scelta di quali caselle saranno quelle numerate, ovvero

quelle da cui le parole hanno origine, e di inserire le definizioni relative. Ciò è realizzato con l'ausilio dei seguenti componenti: per prima cosa un Check Box da spuntare per rendere numerata l'ultima casella selezionata (inizialmente esso risulterà deselezionato per ogni casella che compone il cruciverba), quindi due campi di testo nei quali inserire rispettivamente le definizioni orizzontale e verticale associate alle parole che hanno origine dalla casella attualmente selezionata ed infine un pulsante per salvare le definizioni digitate che riporta la dicitura "Conferma definizioni". Da notare che i due campi di testo ed il pulsante appena citati sono inizialmente disattivati fintanto che il Check Box non viene selezionato, il che significa che l'utente non può interagirvi. Infine, ulteriori elementi che vanno ad arricchire il pannello in analisi sono delle etichette di testo che descrivono i controlli appena citati (tra cui i due campi di testo, con stringhe che riportano "Definizione orizzontale" e "Definizione verticale"), una che notifica il corretto salvataggio delle due definizioni alla pressione del pulsante apposito, ed altre che, vista la presenza di ulteriore spazio inutilizzato, vengono utilizzate per dare rapide istruzioni circa l'utilizzo dei controlli all'interno del pannello stesso e di altri riguardanti altre funzionalità messe a disposizione dall'applicazione e che verranno discusse in seguito;

- **Pannello nord (introduzione):** Il pannello che sta al di sopra di tutti gli altri è lungo e sottile ed ha scopi esclusivamente estetici. Tutto ciò che esso contiene è infatti una singola etichetta di testo contenente un messaggio introduttivo di benvenuto nell'applicazione;
- **Pannello sud (esecuzione):** Il pannello nella parte bassa della schermata dell'applicazione ha dimensioni simili a quello con orientamento nord appena visto, ma il suo contenuto annovera controlli di importanza fondamentale. Vi si trova infatti un campo di testo in cui si richiede di digitare il nome che si vuole assegnare al cruciverba che si sta costruendo, un pulsante che riporta l'etichetta "Genera cruciverba" con il compito di avviare la generazione dell'applicazione Flash finale a costruzione del cruciverba ultimata ed un altro pulsante per la chiusura dell'applicazione, con la dicitura "Esci". Vi è da aggiungere soltanto un'etichetta che va ad indicare ciò che è necessario digitare nel campo di testo, ovvero un nome da assegnare al cruciverba.

Approntata l'interfaccia grafica, possiamo ad analizzare le funzionalità che sono state ideate e che l'applicazione deve mettere a disposizione per consentire la creazione di un cruciverba:

- L'utente deve poter digitare una ed una sola lettera all'interno di ogni casella selezionandola. La casella al momento selezionata viene colorata di giallo per permettere di essere identificata immediatamente.
- Similmente a quanto accadeva nel cruciverba in Flash, si vuole permettere all'utente di digitare parole intere senza il bisogno di spostarsi tra una casella e l'altra tra la digitazione di una lettera e l'altra, effettuando uno spostamento automatico verso destra o verso il basso a seconda della direzione di scrittura attiva. Di conseguenza deve anche essere possibile invertire orientamento di scrittura, e si è pensato di effettuare ciò in risposta alla pressione della combinazione di tasti "Control" più una freccia direzionale, esattamente come avveniva nell'applicazione Flash.
- Deve essere possibile annerire le caselle del cruciverba ed è stato scelto di realizzare ciò tramite la semplice pressione del tasto destro del mouse su di esse.
- L'utente può cambiare le dimensioni del cruciverba in qualsiasi momento tramite i Combo Box appositi potendone vedere i risultati grafici immediati, rappresentati dalla disabilitazione delle caselle in eccesso (o la loro riabilitazione) che vengono oscurate e rese inutilizzabili.
- L'utente può scegliere quali caselle devono essere numerate, ovvero in quali hanno origine una o due parole con delle definizioni associate. Per rendere una casella numerata, è sufficiente selezionare il Check Box apposito mentre la casella in questione è selezionata (colorata di giallo). Le caselle numerate verranno colorate in azzurro, e quindi riconoscibili.
- Deve essere possibile digitare la o le definizioni orizzontale e verticale per ogni casella che è stata resa numerata in precedenza. Ciò può essere fatto tramite i due campi di testo appositi ed infine premendo il pulsante per il salvataggio del testo scritto.
- L'utente deve poter scegliere un nome con il quale riconoscere il cruciverba che crea, digitandolo nello spazio apposito. L'applicazione Flash verrà generata all'interno di una cartella che viene creata appositamente e alla quale viene assegnato il nome scelto.
- Dal menù di cui l'interfaccia grafica è dotata deve essere possibile chiudere l'applicazione, visualizzare una schermata con le istruzioni per l'uso ed una con le informazioni riguardanti l'applicazione stessa.

Per porre l'applicazione in condizione di mettere a disposizione le funzionalità appena riportate sono state costruite delle classi Java contenenti il codice necessario per svolgere tutte le operazioni richieste.

Iniziamo l'analisi di tali classi a partire dal livello più basso dell'architettura *three-tier*: il livello dei dati. Qui è contenuta una classe che sta alla base dell'intera applicazione e che rappresenta l'unità primaria del database; si tratta di *Cell* (ovvero cella in inglese) la quale contiene in sé tutti gli attributi necessari per descrivere una casella del cruciverba.

Ciò che l'applicazione fa è creare tante istanze di questa classe quante sono le caselle di cui è composto il cruciverba che si sta costruendo ed assegnare ad ognuno di questi oggetti i dati che descrivono la casella che essi rappresentano.

L'insieme di tutti questi oggetti viene immagazzinato in una matrice composta da dei vettori, ovvero delle strutture dati indicizzate messe a disposizione ed automaticamente gestite da Java, dalle quali è facile andare a leggere ed ottenere informazioni su uno specifico oggetto in esse contenuto. Tale matrice è composta da un vettore con tanti spazi quante sono le righe che compongono il cruciverba, e dove ognuno di tali spazi contiene a sua volta un altro vettore con tanti spazi quante sono le colonne del cruciverba. In pratica così facendo ogni vettore contenuto nel primo rappresenta una riga di caselle del cruciverba e si viene così ad avere una struttura dati che riflette in astratto l'aspetto della griglia del cruciverba stesso.

Lo scopo fondamentale dell'applicazione è quella di permettere all'utente di inserire in questa struttura dati tutte le informazioni necessarie per la costruzione del cruciverba, infatti la generazione di quest'ultimo avviene basandosi esclusivamente sui dati in essa contenuti al termine della modellazione effettuata dall'utente.

Gli attributi previsti dalla classe *Cell* che quindi descrivono ogni casella sono i seguenti:

- Flag *black*: è una variabile booleana che indica se la casella è annerita (valore *true*) o meno (valore *false*);
- Stringa *id*: è una serie di caratteri che identificano la casella che le varie istanze della classe descrivono. Come per l'attributo *id* già visto all'interno del file XML letto dall'applicazione Flash, questa variabile contiene al suo interno due numeri interi che indicano rispettivamente riga e colonna di appartenenza della casella descritta;
- Carattere *let*: è una variabile composta da un unico carattere il quale contiene la lettera digitata dall'utente in quella specifica casella. Se quest'ultima è annerita, tale variabile non è importante;
- Intero *n*: è un valore intero che contiene il numero che identifica le definizioni visualizzato nelle caselle dalle quali ha origine una parola. Questa variabile assume il valore 0 per le caselle normali, mentre è priva di significato per quelle non numerate o annerite;

- Stringa *ori*: è la stringa in cui viene memorizzata la definizione orizzontale nel caso in cui la casella in esame sia tra quelle numerate e preveda una definizione che segua tale orientamento, altrimenti risulta priva di importanza;
- Stringa *ver*: ha la stessa funzione di *ori*, ma si riferisce alla definizione che segue l'orientamento verticale;
- Intero *n_ori*: è un valore intero che indica il numero della definizione orizzontale che fa riferimento alla parola di cui la lettera contenuta nella casella in esame fa parte. Per fare un esempio prendiamo le prime cinque caselle orizzontali del cruciverba ed immaginiamo che esse compongano una parola di cinque lettere che ha origine nella prima; in quel caso la prima casella sarà sicuramente numerata (in particolare le verrà associato il numero 1), mentre è possibile che le altre non lo siano (a meno che non originino a loro volta una parola in senso opposto, ovvero verticale), ma indipendentemente da ciò la variabile *n_ori* avrà il valore 1 per ognuna delle cinque caselle prese in esame, poiché tutte contengono una lettera che va a comporre la parola alla quale viene associata la definizione orizzontale numero uno. Questa variabile assume valore 0 nel caso in cui la lettera nella casella interessata non faccia parte di una parola orientata orizzontalmente, inoltre è priva di importanza per le caselle annerite;
- Intero *n_ver*: è il corrispettivo di *n_ori*, ma si riferisce alla definizione verticale.

Con questi dati che descrivono ogni casella si hanno a disposizione tutte le informazioni necessarie per la generazione del documento XML così come è stato analizzato in precedenza. Ma come si passa dalla visualizzazione dell'interfaccia grafica alla raccolta di tutti i dati necessari fino alla creazione del cruciverba Flash completo? In altre parole come funziona l'applicazione?

A questo scopo è utile analizzare ogni singolo aspetto tecnico di questo software partendo dalle varie operazioni che l'utente può effettuare interagendo con l'interfaccia grafica, osservando come i gestori degli eventi di cui sono dotati i componenti dell'interfaccia grafica reagiscono a tali interazioni richiedendo i servizi del *logic tier*, ovvero il secondo livello dell'architettura *three-tier*, che fornisce tutti gli strumenti per la gestione e l'elaborazione dei dati tramite la classe *Manager*.

Innanzitutto deve essere osservato come proprio la classe appena citata, *Manager*, crei la matrice che svolge il ruolo di database immediatamente all'avvio dell'applicazione, prima ancora della visualizzazione dell'interfaccia grafica e quindi di qualsiasi interazione con l'utente.

Questa operazione preliminare viene effettuata prendendo in considerazione due particolari: il primo è che i vettori che costituiscono la matrice vengono creati in maniera tale da essere dotati di tanti spazi quanto specificato dalle dimensioni massime previste per il cruciverba (16 per 12),

questo perché all'avvio dell'applicazione tutte le caselle sono attive ed utilizzabili; il secondo riguarda le singole caselle: come è stato detto ognuna di esse viene rappresentata da un'istanza della classe *Cell* ed inserita nella giusta posizione all'interno della matrice, ma la cosa da notare riguarda il valore assegnato inizialmente alle variabili contenute negli oggetti. Essi infatti assumono i seguenti valori: l'identificatore *id* assume già il valore definitivo che identifica la relativa casella e che determina la sua posizione anche all'interno della matrice; la variabile *black* indicherà che la casella è normale (non annerita); *let*, che deve contenere la lettera digitata nella casella, dovrebbe rimanere vuota, per cui le viene assegnato un valore privo di significato (in particolare il carattere asterisco '*'); le tre variabili numeriche che descrivono le definizioni (*n*, *n_ori* e *n_ver*) vengono poste al valore iniziale di 0, il che sta ad indicare l'assenza di definizioni in tale casella; le due stringhe con la funzione di memorizzare le due definizioni che possono essere associate ad una casella (*ori* e *ver*) rimangono per il momento prive di contenuto.

Una volta terminata la generazione dell'intera matrice, l'applicazione si occupa di visualizzare l'interfaccia grafica descritta in precedenza. Osserviamo in dettaglio il funzionamento di tutte le operazioni che avvengono in seguito all'interazione dell'utente.

- **Annerire le caselle:** quando l'utente clicca con un tasto del mouse su una qualsiasi casella che compone il cruciverba, il gestore degli eventi di quest'ultima avvia l'esecuzione di un metodo che svolge una serie di operazioni. Per prima cosa esso rileva la casella da cui l'evento ha avuto origine e quale tasto del mouse lo ha attivato. Se si tratta di quello destro (ovvero l'unico che deve essere in grado di permettere l'annerimento di una casella) allora viene analizzato lo stato attuale di della casella cliccata. Nel caso in cui essa fosse normale allora il metodo si occupa di rimuovere il testo in essa contenuto, di colorarla di nero, di disabilitarla (ovvero renderla non selezionabile e quindi impossibilitare l'utente a digitarvi caratteri) ed infine di richiedere alla classe *Manager* di modificare di conseguenza il database. Nel caso in cui invece la casella fosse già stata precedentemente annerita verranno svolte le azioni contrarie, quindi essa tornerà di colore bianco e verrà riabilitata. Anche in questo caso verrà richiesto a *Manager* di rendere effettiva la modifica anche nel database. Tale richiesta avrà la seguente forma: viene richiesta l'esecuzione del metodo adatto tra quelli messi a disposizione da *Manager* passando ad esso due dati, ovvero l'identificatore della casella interessata ed un valore booleano che sta indicare se essa deve essere memorizzata come "annerita" o meno. Tale metodo risale alla posizione corretta all'interno della matrice in cui si trova l'oggetto relativo alla casella su cui si sta operando tramite il

primo dei due dati, mentre il secondo definisce il valore che verrà assegnato all'attributo *black* dell'oggetto appena ricercato.

- **Cambiare le dimensioni del cruciverba:** nel momento in cui l'utente modifica i valori selezionati tramite i due Combo Box che indicano le dimensioni rispettivamente di larghezza ed altezza, indipendentemente che venga modificato l'uno o l'altro, il gestore degli eventi esegue un metodo che svolge tutte le operazioni necessarie perché il cambiamento abbia effetto. Per prima cosa vengono letti dai due Combo Box i nuovi rispettivi valori di larghezza ed altezza, quindi, conoscendo essi, si controllano una per una tutte le caselle del cruciverba nell'interfaccia grafica ed in particolare i loro identificatori. Se viene rilevata una casella posizionata in una colonna che sta oltre il valore di larghezza scelto o in una riga oltre il valore di altezza, allora essa viene disabilitata (come avveniva per l'annerimento) e colorata di grigio. Al contrario, se viene trovata una casella disabilitata e grigia che sta all'interno delle dimensioni scelte, essa viene riabilitata e resa nuovamente bianca. Una volta completata questa operazione viene richiesto a *Manager* di fare la stessa cosa nell'ambito del database richiamando un suo metodo apposito e comunicandogli i due valori scelti per le dimensioni. Tale metodo di *Manager* si occupa per prima dell'altezza: se essa viene ridotta verranno semplicemente rimossi tanti vettori riga dalla matrice quanti necessari per accordarsi con la nuova altezza, mentre invece ne saranno inseriti di nuovi nel caso in cui si debba aumentarne il numero (in quest'ultimo caso ogni spazio delle nuove righe viene riempito con istanze della classe *Cell* create con le stesse modalità di quanto già fatto alla prima costruzione del database all'avvio dell'applicazione). Viene poi ripetuto il procedimento visto per le modifiche alla larghezza; in questo caso però non è più sufficiente aggiungere o rimuovere vettori riga, ma è necessario sfogliarli tutti rimuovendovi o aggiungendovi in coda dei nuovi spazi, nuovamente riempiti con istanze di *Cell* come già visto.
- **Selezionare una casella:** come già visto nell'applicazione Flash, nel momento in cui una casella del cruciverba viene selezionata, viene attivato un evento di acquisizione del focus da parte della casella stessa. Il gestore di tale evento si occupa di eseguire un metodo che si occupa di una serie di operazioni necessarie ogniqualvolta ciò avvenga: per prima cosa viene identificata la casella appena selezionata, quindi viene interrogato *Manager* per sapere se essa era stata in precedenza numerata o meno tramite l'esecuzione di un metodo da esso messo a disposizione in grado di ottenere tale informazione dal database a partire dall'identificatore della casella interessata. A questo punto nel caso in cui la casella non fosse numerata, il metodo disabilita i controlli relativi all'inserimento delle definizioni (i due

campi di testo appositi ed il pulsante per la loro memorizzazione) e deselecta il Check Box che sta proprio ad indicare se la casella corrente è numerata o meno. Nel caso contrario le operazioni che vengono effettuate prevedono esattamente l'opposto, ovvero i controlli che nel caso precedente venivano disabilitati ora vengono abilitati, ed inoltre si interroga nuovamente *Manager* per sapere se in precedenza si erano già salvate delle definizioni associate alla casella in esame. Ciò si ottiene in maniera molto simile alla precedente richiesta, infatti *Manager* possiede un metodo in grado di ottenere tale dato dal database conoscendo l'identificatore della casella interessata. Chiaramente nel caso in cui delle definizioni siano già presenti, queste vengono stampate nei due campi di testo appositi in modo tale che l'utente le possa visualizzare e modificare. L'ultima operazione da effettuare è gestire la colorazione delle caselle: viene effettuata una ricerca nell'intero cruciverba per trovare la casella selezionata in precedenza (sarà colorata in giallo); se ne viene trovata una (è possibile che non ve ne siano, ad esempio se l'applicazione è appena stata avviata e non vi erano caselle selezionate in precedenza), questa verrà riportata alla colorazione normale bianca oppure azzurra nel caso sia una casella numerata, infine la colorazione gialla viene assegnata a quella attuale indicandone la selezione. Viene registrato in un flag se la sua colorazione precedente era bianca o azzurra, in modo da essere in grado di ripristinare il colore corretto nel momento in cui verrà effettuata la successiva selezione.

- **Assegnare o rimuovere un numero di definizione da una casella:** il controllo con il quale l'utente deve interagire per attivare tale funzionalità è il Check Box, il quale se selezionato indica la casella come numerata e come normale in caso contrario. Quando l'utente interagisce con esso, il gestore degli eventi ad esso associato esegue un metodo che svolge le seguenti operazioni: per prima cosa osserva se l'interazione ha comportato la selezione o la deselecta del Check Box, quindi nel primo caso si occupa di abilitare i campi di testo per la scrittura delle definizioni ed il pulsante per la loro memorizzazione, di colorare la casella in azzurro (colore che permette di riconoscere quelle numerate da tutte le altre) ed infine di richiedere a *Manager* di inserire l'informazione nel database organizzando la numerazione di tutte le caselle. Nel secondo caso chiaramente i tre componenti con lo scopo di gestire le definizioni (i due campi di testo ed il pulsante) vengono disabilitati rimuovendo l'eventuale testo precedentemente digitato nei campi di testo, la colorazione della casella viene reimpostata bianca e nuovamente viene richiesto a *Manager* di gestire le modifiche da effettuare nel database. Il metodo messo a disposizione da *Manager* per rispondere a tali richieste necessita di due informazioni per procedere, cioè l'identificatore della casella su cui si sta operando ed un flag che indichi se essa è stata appena indicata come "da

numerare” o il contrario. Sapendo ciò tale metodo individua la casella interessata all’interno della matrice e, nel caso la debba rendere numerata, pone la sua variabile n al valore temporaneo 1, mentre nel caso contrario esso pone a 0 non solo n , ma anche n_ori e n_ver , inoltre rimuove il contenuto delle stringhe ori e ver perdendolo definitivamente. Dopo aver fatto ciò, in entrambi i casi il metodo risistema l’intera numerazione di tutte le caselle, infatti aggiungendone o rimuovendone una, quelle in posizione successiva all’interno del cruciverba e del database devono essere aggiornate di conseguenza. Per fare un esempio basti immaginare un cruciverba con due caselle numerate nella prima riga e due nell’ultima, con numerazione crescente che andrà quindi da 1 a 4. Poniamo di numerare una casella tra quelle della riga centrale; ad essa dovrà essere associato il numero 3 e di conseguenza le due in ultima riga passeranno a valori pari a 4 e 5. Questa operazione è svolta sfogliando ordinatamente tutte le caselle nel database e, per quelle tra esse con la variabile n che presenta un valore diverso da 0 (il che significa che sono numerate), tale valore viene sostituito con un numero crescente partendo da 1, quindi alla prima verrà associato proprio il valore 1, alla seconda 2 e via dicendo. Lo stesso valore viene assegnato non solo ad n ma anche a n_ori ed n_ver nel caso anche esse avessero presentato un valore diverso da 0. Ulteriori chiarimenti circa l’utilità di queste ultime variabili verranno trattati successivamente, per ora basta ricordare che ad esse viene assegnato il valore definitivo utile per la generazione del cruciverba vero e proprio solo prima dell’effettiva creazione dell’applicazione Flash.

- **Digitare e salvare le definizioni per le caselle numerate:** dopo aver digitato le definizioni (oppure una sola delle due) negli appositi campi di testo, l’utente deve premere il pulsante che reca la dicitura “Conferma definizioni” per far sì che esse vengano realmente salvate all’interno del database. Ciò che accade alla pressione del pulsante è l’esecuzione di un metodo da parte del gestore degli eventi che svolge solo due operazioni: la prima è una richiesta a *Manager* per l’effettiva memorizzazione delle definizioni, l’altra è la stampa di un messaggio di conferma in un’apposita etichetta di testo sottostante il pulsante stesso, il quale sarà rimosso quando l’utente continuerà la costruzione del cruciverba ed in particolare alla successiva selezione di un’altra casella. A *Manager* vengono comunicati l’identificatore della casella interessata e le due stringhe contenenti le definizioni digitate dall’utente; sapendo ciò il metodo in esso contenuto approntato per eseguire tale memorizzazione svolgerà le seguenti operazioni: per prima cosa individua la casella corretta all’interno della matrice, quindi setta le sue due stringhe ori e ver ai valori ottenuti dalla richiesta dell’interfaccia grafica (se una delle due definizioni è stata lasciata vuota, nella variabile

associata ad essa viene inserito per convenzione solo un carattere asterisco ‘*’) ed infine fa assumere alle variabili n_ori e n_ver lo stesso valore contenuto in n (cioè solo nel caso in cui la rispettiva definizione non sia vuota, nel qual caso la relativa variabile rimane invariata con valore 0).

- **Digitare lettere nelle caselle, spostarsi tra di esse e cambiare direzione di scrittura:** tutte queste operazioni messe a disposizione dell’utente sono rese disponibili dal gestore degli eventi associato alle varie caselle in grado di rilevare la pressione dei tasti battuti alla tastiera mentre una di esse è selezionata. A seconda di quale è stato premuto vengono intraprese operazioni diverse, in maniera molto simile a quanto visto per l’applicazione Flash: se il tasto rilevato è “Control” oppure “Backspace” per cancellare una lettera, allora non verrà eseguita alcuna ulteriore operazione; se invece viene rilevata la pressione prolungata del tasto “Control” in combinazione con una qualsiasi delle frecce direzionali, allora verrà invertita la direzione di scrittura (come accadeva per l’applicazione Flash, questa viene indicata con una variabile apposita di tipo booleano che acquisisce il valore “false” in caso di scrittura orizzontale e “true” in caso contrario; al verificarsi dell’evento appena descritto il valore da essa attualmente presentato viene cambiato nel suo opposto); nel caso si tratti di un altro tasto si procede innanzitutto stampando la lettera associata ad esso (se di una lettera si tratta) con formattazione maiuscola nella casella in cui si sta operando, quindi si assegna il focus ad un’altra casella adiacente a seconda dei seguenti fattori: se è stata premuta una delle quattro frecce direzionali viene acquisito il focus dalla casella adiacente nella direzione specificata dalla freccia stessa, se invece il tasto premuto è un altro ci si sposta in quella a destra nel caso in cui l’orientamento della scrittura attivo in quel momento sia orizzontale, altrimenti in quella sottostante. In conclusione è da notare che se il metodo appena analizzato porta ad uno spostamento tra caselle, quest’ultima operazione determina a sua volta la selezione di una nuova casella con tutte le conseguenze derivanti da ciò, viste in precedenza;
- **Chiudere l’applicazione:** premendo l’apposito pulsante l’utente può chiudere l’applicazione in qualsiasi istante. L’unica operazione svolta dal metodo richiamato dal gestore degli eventi al presentarsi di questa interazione è proprio quella di terminare l’esecuzione del programma. Lo stesso effetto può essere ottenuto anche dalla voce “Esci” nel menù denominato “File” o semplicemente cliccando sulla “X” della finestra in cui appare l’applicazione.
- **Generare il cruciverba in Flash:** quando l’utente termina ogni aspetto della costruzione del cruciverba può premere il pulsante “Genera cruciverba”, il quale da avvio ad un processo

che ha come risultato finale la creazione dell'applicazione Flash pronta per l'uso. Osserviamo tutti i passaggi necessari per ottenere tale prodotto: la prima cosa che il metodo eseguito dal gestore degli eventi fa è verificare che l'utente abbia digitato un nome per il cruciverba che si sta costruendo nell'apposito campo di testo. Nel caso ciò non sia stato fatto, viene visualizzato un *Dialog*, ovvero una piccola finestra, contenente un messaggio che informa di tale mancanza, ed il processo viene interrotto per la mancanza di tale dato; nel caso invece in cui venga rilevata una stringa il processo può continuare con la verifica di tutte le caselle che compongono il cruciverba, ed in particolare con il controllo che ognuna di esse (con l'esclusione di quelle disabilitate, ovvero annerite oppure fuori dalle dimensioni scelte per il cruciverba) contenga esattamente un carattere, né più né meno. Se ciò non viene verificato l'utente ottiene di nuovo un *Dialog* contenente un messaggio di notifica ed il processo viene interrotto; in caso contrario invece si prosegue con le operazioni necessarie. Il passo successivo è la creazione di un vettore in cui per ogni casella abilitata viene memorizzata una stringa contenente il suo identificatore e la lettera in essa digitata, quindi tale vettore viene comunicato a *Manager* in modo tale che esso possa occuparsi della memorizzazione delle lettere associate ad ogni casella nel database. Il metodo di *Manager* incaricato di fare ciò non si limita però ad individuare le varie caselle all'interno della matrice e memorizzarvi la lettera assegnatagli, ma svolge anche un'ulteriore operazione, ovvero la verifica che per ogni casella numerata tra tutte quelle in database sia stata memorizzata almeno una delle due definizioni ad esse associate. Se vengono rilevate delle definizioni mancanti ciò sarà notificato all'utente tramite un *Dialog* contenente un messaggio esplicativo ed anche in questo caso il processo verrà interrotto, altrimenti le operazioni possono proseguire con un'altra richiesta rivolta alla classe *Manager*: l'assegnazione delle definizioni a tutte le caselle, anche a quelle non numerate, in modo che ad ognuna vengano associate le due (o in alcuni casi una sola) definizioni che descrivono le due parole di cui la lettera nelle caselle stesse fa parte. Per ottenere tale risultato semplicemente si sfogliano una per una tutte le caselle del cruciverba e, se le sue variabili n_{ori} ed n_{ver} presentano valori uguali a 0, sostituendo tale valore con quello che detengono le stesse variabili della casella precedente (chiaramente quella a sinistra nel caso di n_{ori} e quella sovrastante nel caso di n_{ver}) e lo stesso dicasi per le stringhe *ori* e *ver* contenenti le due definizioni. Facciamo un esempio prendendo in esame la seconda casella del cruciverba, immaginando che alla prima siano state associate entrambe le definizioni (quindi sia n_{ori} che n_{ver} possiedono un valore diverso da 0, in questo caso 1 visto che si tratta della prima casella) mentre alla seconda soltanto quella verticale (quindi n_{ver} avrà

valore 2, mentre *n_ori* sarà rimasta uguale a 0). Con l'esecuzione del metodo che si sta analizzando ciò che accade è che viene rilevato il valore 0 che la variabile *n_ori* della seconda casella presenta, quindi esso viene sostituito con quello della casella precedente in senso orizzontale, quindi la prima, così da farlo diventare 1. Allo stesso modo la variabile *ori* della seconda casella contenente la definizione ad essa associata con orientamento orizzontale ma fino a quel momento rimasta vuota, verrà riempita con la stessa stringa contenuta nella variabile *ori* della casella precedente. In tal modo, nel momento in cui verrà costruita l'applicazione Flash, si potranno visualizzare le definizioni attive in entrambe le direzioni qualsiasi sia la casella selezionata. Svoltata questa operazione è il momento dell'avvio del vero e proprio processo di creazione dell'applicazione Flash ora che la matrice contenente tutti i dati necessari è completa. In fase di sviluppo si è optato per inserire tutto il codice utile per la fase della generazione in una classe a parte che è stata chiamata *Generator*, anch'essa facente parte del secondo livello dell'architettura *three-tier*, ovvero il *logic tier*, sicché, giunti a questo punto, ciò che *Manager* fa è creare un'istanza di tale nuova classe per usufruire delle funzionalità da essa offerte, e nel fare ciò comunica ad essa i dati di cui si avrà bisogno per la costruzione effettiva del cruciverba in Flash, ovvero il nome che è stato ad esso assegnato dall'utente, le sue dimensioni e la matrice contenente le informazioni su ogni casella che lo compone. Una volta resa disponibile tale istanza, *Manager* prosegue nel processo appoggiandosi sui metodi forniti da essa, iniziando dal controllo per verificare che una cartella con il nome scelto dall'utente per il cruciverba non sia già esistente. Se *Generator* non rileva alcuna directory con tale nome il processo continua, mentre in caso contrario nuovamente un *Dialog* comunica all'utente il problema e l'interruzione delle operazioni. La successiva operazione che *Manager* richiede alla classe generatrice è la creazione della cartella che si è appena verificato non esistere ancora e, all'interno di essa, dei due file dei quali verrà successivamente prodotto il contenuto, ovvero il file XML di cui si è già ampiamente discusso e la pagina HTML che ospiterà l'oggetto Flash dell'applicazione Gaupe che andrà a leggere il documento XML. Ancora una volta se l'operazione va a buon fine il processo può continuare la sua esecuzione, altrimenti l'utente vedrà visualizzato un *Dialog* che lo informa sul problema in cui ci si è imbattuti. A questo punto *Manager* invia la sua ultima richiesta a *Generator*, ovvero la scrittura di tutti i file necessari per il corretto funzionamento dell'applicazione Flash, che, come già detto in precedenza, non sono soltanto i due documenti appena citati, ma anche ad esempio alcuni oggetti Flash che costituiscono l'applicazione Gaupe. Tutti questi file non richiedono alcuna modifica al variare del cruciverba che contribuiranno a visualizzare, quindi la soluzione

ideata è semplicemente quella di copiarli nella cartella che è stata appena creata in cui verrà costruita l'applicazione Flash. Ovviamente per fare ciò è necessaria una locazione da cui reperire tali file, quindi si è resa necessaria un'ulteriore directory (denominata “resources”) posta nella stessa posizione da cui l'applicazione Java viene eseguita contenente tutte queste risorse fisse necessarie. Appurata la presenza di tale cartella con il suo essenziale contenuto, è possibile tornare ad analizzare il modo in cui la classe *Generator* procede per ottenere il risultato finale perfettamente funzionante. Per prima cosa questa classe generatrice si occupa proprio della copia dei file appena citati all'interno della cartella creata appositamente per il cruciverba che si sta costruendo, e per fare ciò ha a disposizione un metodo ricorsivo che opera in questo modo: inizia con l'analisi della prima delle due locazioni che gli vengono comunicate (si tratta dell'origine, ovvero la cartella *resources*, e della destinazione, cioè la cartella approntata per ospitare il nuovo cruciverba), rilevando tutti i file in essa contenuti. Per ognuno di essi si comporta in maniera differente a seconda che si tratti di file veri e propri o di sottocartelle (*resources* ne contiene una contenente a sua volta altri file necessari per l'applicazione), infatti per ogni file rilevato eseguirà una copia byte per byte dalla locazione d'origine a quella di destinazione, mentre per ogni directory rilevata esso ne costruisce una con lo stesso nome nella destinazione per poi richiamare se stesso con diverse informazioni circa le due locazioni, avendo come origine la directory in analisi di cui si è appena costruita una copia, mentre come destinazione proprio tale copia. In tal modo il processo di copia viene eseguito ricorsivamente a cascata per l'intero contenuto dell'originaria cartella di origine, ovvero *resources*. Fatto ciò è finalmente il momento di elaborare il contenuto dei due file precedentemente aperti, l'XML e la pagina HTML: per la seconda non vi sono particolari accorgimenti da riportare poiché non vi sono parti dipendenti dal cruciverba che si sta creando; la si genera e non la si copia come avviene per tutti gli altri file visti poc'anzi esclusivamente per permetterne un'eventuale personalizzazione tramite la modifica del codice dell'applicazione Java. Per quanto riguarda il file XML invece la sua costruzione è stata già ampiamente commentata; esso viene costruito sulla base dei dati contenuti nella matrice raccolti durante la modellazione del cruciverba, avendo quindi a disposizione informazioni come lo stato delle singole caselle (normali, annerite o numerate), il loro eventuale numero che identifica le definizioni, la lettera corretta che devono ospitare e le due definizioni ad esse associate. Oltre a ciò è necessario avere informazioni anche sulle dimensioni del cruciverba, dati ottenuti al momento della creazione dell'istanza di *Generator* da parte di *Manager*, per permettere al metodo generatore di calcolare con precisione le coordinate da assegnare ai vari componenti

grafici che andranno a comporre l'applicazione Flash; in particolare con la misura della larghezza è possibile risalire alla corretta posizione delle singole caselle in una riga del cruciverba con la tecnica illustrata nella sezione in cui si descrive il contenuto del file XML, mentre l'altezza rappresenta la misura di partenza oltre la quale è possibile disegnare gli elementi dell'interfaccia grafica di controllo dell'applicazione Flash senza il rischio di raffigurarli a delle coordinate occupate dalla griglia del cruciverba stesso. Infine, una volta terminata la scrittura di questi due ultimi file, l'applicazione Flash può dirsi terminata e pronta all'utilizzo, quindi, terminata l'opera della classe *Generator*, verrà visualizzato un *Dialog* che notifica con un messaggio il termine delle operazioni di creazione;

- **Visualizzare le istruzioni per l'uso:** navigando nel menù "Help" di cui l'interfaccia grafica dell'applicazione è dotata l'utente ha la possibilità di leggere le istruzioni per l'uso delle varie funzionalità del programma. In tale menù si trova infatti la voce omonima "Help" che, se cliccata dall'utente, porta all'esecuzione di un metodo, lanciato dal gestore degli eventi, in grado di costruire e visualizzare una finestra contenente un'area di testo riportante l'intero testo che compone le istruzioni (navigabile tramite un apposito scroll) oltre ad un pulsante per la chiusura della finestra al termine della lettura;
- **Visualizzare informazioni sull'applicazione:** dal menù "Help" è possibile accedere anche ad una schermata che mostra delle informazioni basilari sull'applicazione; è infatti presente la voce "About" il cui gestore si comporta in maniera molto simile a quello della precedente voce vista, ovvero al click dell'utente su di essa esegue un metodo che si occupa della costruzione e della visualizzazione di un'altra finestra molto simile a quella contenente le istruzioni, al posto delle quali però in questo caso rende disponibili informazioni quali l'autore dell'applicazione, l'ambito di sviluppo e la data.

L'applicazione Java generatrice di cruciverba in Flash si può a questo punto dire ultimata e funzionante, ma sono prontamente state pensate delle migliorie e delle funzionalità da applicarvi.

Due in particolare sono le cose che si è pensato di aggiungere: la prima consiste in un pulsante che implementi la stessa funzionalità ottenuta tramite la pressione della combinazione di tasti "Control" più una delle frecce direzionali, ovvero il cambio della direzione di scrittura, operazione precedentemente irraggiungibile tramite i controlli dell'interfaccia grafica, accompagnato dall'introduzione di un'etichetta contenente del testo che indichi esplicitamente quale delle due direzioni è attiva al momento, così come accade nell'applicazione Flash; la seconda invece riguarda l'introduzione di una nuova funzione, ovvero il suggerimento delle parole da inserire nel cruciverba tramite l'uso di un dizionario, o meglio, una lista di parole.

Si è pensato di realizzare ciò tramite la semplice introduzione di un pulsante alla pressione del quale l'applicazione rileva la casella selezionata attualmente e quelle adiacenti che andrebbero a costituire una parola, le eventuali lettere già in esse digitate ed il senso di scrittura attivo al momento. Partendo da tali informazioni il programma deve essere in grado di filtrare tutte le parole contenute nel dizionario che si sceglie di utilizzare fino a trovare quelle che si adatterebbero agli spazi disponibili per la parola (lunghezza della parola stessa) ed alle lettere già digitate nelle caselle che compongono la parola che si sta ricercando.

Per fare un esempio poniamo di aver selezionato la terza casella di una serie orizzontale composta da cinque di esse con alle estremità due caselle annerite che le delimitano; inoltre nella seconda casella è già stata digitata la lettera 'M', nell'ultima la lettera 'E', mentre le altre tre sono ancora vuote. Supponendo che la direzione di scrittura attiva al momento è proprio quella orizzontale, premendo il pulsante per il suggerimento delle parole da dizionario nello stato appena descritto, il risultato sarà una finestra con elencate tutte le parole presenti nel dizionario che soddisfino i seguenti requisiti:

- Devono essere lunghe non più di cinque lettere (si è scelto di fornire anche parole più brevi, in quanto sarebbe in seguito semplice rimodellare il cruciverba annerendo delle caselle in modo da renderlo adatto ad ospitare una parola più breve di quanto originariamente previsto);
- La seconda loro lettera deve essere una 'M';
- La quinta loro lettera deve essere una 'E', ma quest'ultimo requisito riguarda solo le parole composte da cinque lettere, non quelle più brevi.

Di conseguenza l'applicazione dovrebbe stampare in una finestra parole come ad esempio "amore", ma anche più brevi come "amo".

Come già detto entrambe queste due nuove funzionalità prevedono l'introduzione di un nuovo pulsante nell'interfaccia grafica dell'applicazione: è stata per loro scelta una posizione sottostante la griglia del cruciverba piuttosto che alla sua destra come tutti gli altri controlli, in modo da non rendere troppo pesanti e poco immediati l'utilizzo e l'interazione con tale area.

Facendo riferimento alla struttura a quattro pannelli dell'interfaccia grafica dell'applicazione, i due nuovi controlli vengono posti nella parte inferiore del pannello ovest, ovvero quello che in precedenza ospitava esclusivamente il cruciverba vero e proprio.

Analizziamo dunque in dettaglio anche queste due nuove funzionalità offerte dall'applicazione sviluppata:

- **Cambiare la direzione di scrittura:** alla pressione da parte dell'utente del pulsante "Cambia" appena aggiunto all'interfaccia grafica viene avviato dal gestore degli eventi un metodo con il compito di svolgere le stesse operazioni eseguite in caso di pressione della combinazione di tasti "Control" più freccia direzionale, ovvero semplicemente il cambiamento del valore della variabile booleana che rappresenta la direzione in uso da "false" (che sta ad indicare il senso orizzontale) a "true" (che indica quello verticale) o viceversa a seconda del valore corrente da essa assunto. Vi è solo una differenza: ora è necessario modificare anche il testo contenuto nell'etichetta che accompagna il pulsante, la quale indica esplicitamente il senso di scrittura attivo, chiaramente in funzione della variabile booleana che lo descrive. Da notare che quest'ultima operazione deve essere svolta anche con la pressione della combinazione di tasti, quindi è risultato necessario estendere il codice eseguito al presentarsi di tale evento, rendendo anch'esso in grado di modificare il testo contenuto nell'etichetta;
- **Visualizzare parole da inserire nel cruciverba suggerite da un dizionario:** prima di descrivere in dettaglio questa funzionalità è necessario notare che si è preferito creare una nuova classe, che va ad aggiungersi a quelle che già compongono l'applicazione, contenente tutti i metodi e le strutture necessarie alla gestione ed all'uso del dizionario. Un'istanza di questa classe, denominata *Dictionary*, parte del *logic tier* all'interno dell'architettura *three-tier* dell'applicazione, viene creata all'avvio del programma, prima ancora della visualizzazione dell'interfaccia grafica, ed immediatamente viene costruita una struttura dati, in particolare un vettore, nel quale vengono memorizzate tutte le parole contenute nel file dizionario che viene passato all'applicazione. Essa si aspetta di leggere un file contenente un parola per ogni riga; se il file dizionario è così formattato la lettura verrà svolta senza alcun problema. È stato ricercato in rete un dizionario così strutturato ed il più completo ed esaustivo possibile; infine ne è stato scelto uno proposto da Gilda Anacronisti, una comunità virtuale per l'intrattenimento intelligente, sul suo sito internet¹⁸. Come viene riportato, esso è stato tra l'altro utilizzato fino al 2005 dall'Internet Scrabble Club¹⁹, sul cui sito internet è possibile giocare con l'omonimo gioco in diverse lingue, tra cui appunto l'italiano. Veniamo ora ad analizzare ciò che avviene alla pressione da parte dell'utente del pulsante "Suggerisci dal dizionario": il metodo eseguito dal gestore degli eventi si occupa

¹⁸ Vedere il sito internet della Gilda Anacronisti (<http://gilda.it/>) ed in particolare la pagina riservata agli elenchi di parole: <http://gilda.it/giochidiparole/elenchi.htm>

¹⁹ Vedere il sito internet dell'Internet Scrabble Club: <http://www.isc.ro/>

innanzitutto di rilevare la casella selezionata al momento di tale interazione con l'utente, quindi passa a sfogliare le caselle a ritroso verso sinistra o verso l'alto, a seconda della direzione di scrittura attiva, fino a arrivare alla prima che compone la parola di cui la casella selezionata fa parte. A questo punto viene inizializzata una stringa in cui verrà salvata la parola composta dalle lettere contenute nella serie di caselle in analisi, sostituendo alle lettere mancanti, ovvero alle caselle vuote, un asterisco '*'. Riprendendo l'esempio fatto in precedenza in cui si prendono in considerazione cinque caselle, delle quali la seconda contiene la lettera 'M' e la quinta la lettera 'E', otterremmo quindi una stringa che si presenterebbe in questo modo: *M**E. Il metodo ora effettua un controllo sulla stringa appena ricostruita, ed in particolare verifica che essa contenga almeno un asterisco, il che significa che essa deve essere incompleta; nel caso in cui la parola sia già completa non ha alcun senso proseguire con la ricerca di parole che consentano di riempire gli spazi vuoti, quindi viene visualizzato un *Dialog* contenente un messaggio che mette l'utente al corrente di ciò, mentre nel caso contrario il processo di ricerca nel dizionario può avere inizio. Per fare ciò viene richiamato un metodo messo a disposizione dalla classe *Dictionary* il quale, avendo la stringa contenente gli asterischi precedentemente costruita, è in grado di filtrare tutte le parole contenute nel dizionario e memorizzare in un vettore solo quelle che si adattano negli spazi a disposizione. Tale filtrazione viene svolta nel seguente modo: viene sfogliata ogni singola parola presente nel dizionario e per ognuna di esse viene innanzitutto verificata la lunghezza che deve essere uguale o minore di quella della stringa da completare; se questo primo test viene superato allora si passa ad analizzare le singole lettere verificando che quelle già presenti nella stringa in analisi devono presentarsi uguali nella stessa posizione delle parole che si stanno filtrando (sempre che esse siano abbastanza lunghe da presentare una lettera in quella posizione). Una volta memorizzate tutte le parole adatte nel vettore appositamente creato, si procede con la costruzione di una finestra molto simile a quelle già citate, mostrate cliccando sulle voci di menù "Help" e "About", composte da un'area di testo in cui viene stampato l'elenco di tali parole e da un pulsante con il solo compito di chiudere tale finestra. Nel caso in cui nessuna parola adatta venga trovata, allora nell'area di testo verrà stampato solo un messaggio che mette l'utente al corrente di ciò.

L'implementazione della funzionalità che rende l'applicazione in grado di fornire dei suggerimenti sulle parole prese da un dizionario con cui riempire il cruciverba che si sta costruendo ha reso necessari degli ulteriori provvedimenti utili per la gestione del dizionario stesso. In particolare si è

pensato di implementare la possibilità di aggiungere parole nuove parole nel dizionario laddove esse non siano già presenti e conseguentemente di ordinarle alfabeticamente.

In dettaglio, tali funzionalità hanno portato a modificare lievemente il processo che viene eseguito alla pressione del pulsante per la generazione finale del cruciverba in Flash aggiungendovi delle nuove operazioni da svolgere con il supporto di metodi appositamente creati messi a disposizione dalle classi *Manager* e *Dictionary*. Si tratta di raccogliere tutte le parole che compongono il cruciverba, di filtrarle in modo tale da rilevare quali tra esse non sono ancora presenti nel dizionario, quindi di aggiungerle al dizionario stesso; infine si riordina alfabeticamente l'intero contenuto del dizionario (incluse quindi le parole appena aggiunte) e lo si riscrive aggiornato sul file dal quale è stato letto.

Tutto ciò viene svolto immediatamente dopo la verifica del corretto completamento del cruciverba e prima che ad ogni casella vengano assegnate le definizioni relative alla parola di cui fa parte e di avviare il processo per la vera e propria scrittura dei file necessari alla costruzione dell'applicazione Flash. Le operazioni appena elencate vengono analizzate qui di seguito:

- **Raccolta delle parole che compongono il cruciverba:** per fare ciò viene richiesto a *Manager* di creare un vettore nel quale inserirle tutte. Esso opera sfogliando tutte le caselle del cruciverba alla ricerca di quelle numerate, ovvero quelle in cui hanno origine una o due parole, e, per ognuna di esse, passa ad analizzare le caselle adiacenti memorizzando le lettere in esse contenute in una stringa fino al completamento di tali parole. Sono le variabili *n_ori* ed *n_ver* che comunicano se da una casella hanno origine due oppure una sola parola ed in quest'ultimo caso in quale delle due direzioni è orientata. Via via che il metodo ricostruisce una parola, essa viene inserita nel vettore richiesto per le successive operazioni, in modo che in esso alla fine vi siano contenute tutte quelle di cui il cruciverba è composto;
- **Inserimento nel dizionario delle parole non ancora presenti in esso:** è la classe *Dictionary* che si occupa di ciò una volta che le viene passato il vettore appena costruito contenente tutte le parole del cruciverba. Ricordiamo che essa è già in possesso di un altro vettore nel quale è elencato l'intero contenuto del dizionario; è quindi sufficiente prendere un elemento per volta dal vettore delle parole del cruciverba e confrontarlo con tutti quelli provenienti dal dizionario alla ricerca di uno uguale, il che indicherebbe la presenza della stessa parola e renderebbe quindi inutile il suo inserimento. Quelle invece per le quali non viene trovata alcuna copia identica vengono semplicemente aggiunte in coda al vettore del dizionario, senza quindi rispettare alcun ordine alfabetico;

- **Ordinamento alfabetico delle parole del dizionario:** altra operazioni di cui si incarica la classe *Dictionary*. Vi sono molti algoritmi di ordinamento ideati e studiati, ognuno con i propri pro e contro, ma per il riordinamento delle parole del dizionario è stato scelto l'utilizzo del popolare *quicksort*, ovvero un algoritmo ricorsivo che opera suddividendo in due parti il vettore da riordinare e richiamando se stesso su entrambe le due metà fino al termine dell'intero processo: un metodo d'azione che prevede la suddivisione del problema in parti più piccole e semplici, detto in gergo *divide et impera*. La prima operazione da esso svolta è la scelta di un elemento del vettore chiamato *pivot*, quindi procede spostando a sinistra di tale elemento tutti quelli che devono precederlo ed alla sua destra tutti quelli che vengono dopo. Le stesse operazioni vengono eseguite nuovamente per entrambe le due metà del vettore e così via, ottenendo alla fine dell'intero processo un vettore perfettamente riordinato. Analizzando l'algoritmo *quicksort* da un punto di vista della complessità computazionale, si nota che il suo svolgimento nel caso peggiore, ovvero quello in cui ad ogni passo viene scelto un elemento *pivot* tale che uno dei due sotto-vettori prodotti risulti vuoto, quindi ad esempio scegliendo l'elemento più piccolo del vettore su cui si opera, preveda di richiamare se stesso ricorsivamente un numero di volte pari al numero di elementi contenuti nel vettore su cui esso opera meno uno, con ad ogni passo un uguale numero di confronti tra elementi, per un costo totale di ordine quadratico. D'altra parte nel caso migliore, ovvero quando ad ogni passo viene scelto un elemento *pivot* tale che il vettore venga diviso in due parti dalle stesse dimensioni, tale costo si riduce all'ordine logaritmico e lo stesso vale per il caso medio, confermando l'algoritmo uno tra i più efficienti nelle giuste condizioni. Non vi è alcuna regola precisa nella scelta dell'elemento *pivot* ad ogni passo dell'algoritmo, perciò nel nostro caso è stato scelto di assegnare tale proprietà all'elemento centrale del vettore;
- **Scrittura del file contenente il dizionario aggiornato:** anche quest'ultima operazione è svolta dalla classe *Dictionary*. Semplicemente essa si occupa di aprire il file di testo dal quale l'applicazione aveva letto il dizionario e di riscriverne l'intero contenuto inserendovi le parole contenute nel vettore ora ordinato alfabeticamente, una per ogni riga.

Con ciò si conclude l'implementazione della prima versione perfettamente funzionante dell'applicazione Java per la generazione di cruciverba in Flash (Fig.4).



Fig.4: Screenshot della prima versione del generatore di cruciverba

Dopo una serie di test e di utilizzi del programma è sorta però la necessità di miglioramenti dal punto di vista sia grafico che operativo, oltre all'ideazione di nuove funzionalità da aggiungere a quelle già messe a disposizione dal programma. L'insieme di tutte queste innovazioni ha portato alla scelta di riprogettare interamente l'applicazione in modo da renderla non solo aggiornata seguendo le linee guida ideate, ma anche maggiormente mantenibile in vista di eventuali ulteriori estensioni future.

5.7. La riprogettazione dell'applicazione Java

Come detto poco sopra è stato l'insieme di modifiche e aggiornamenti ad indirizzare verso la scelta di rielaborare l'intera applicazione, principalmente per la loro quantità non trascurabile, accompagnato dalla necessità di rendere l'intero codice più mantenibile e più adeguatamente modularizzato per una migliore leggibilità e per rendere ulteriori estensioni più facilmente implementabili.

Qui di seguito vengono elencate tutte le modifiche che si è pianificato di applicare:

- Innanzitutto si è deciso di rielaborare interamente l'interfaccia grafica passando dalla presentazione di un'unica finestra nella quale si modella il cruciverba in ogni suo dettaglio a quella di tre finestre consecutive, ognuna con il compito di consentire all'utente di immettere i dati relativi ad un particolare aspetto del cruciverba in maniera più chiara ed intuitiva. In particolare la prima finestra visualizzata consentirebbe la modellazione della griglia che compone il cruciverba, con dettagli quali le dimensioni, la posizione delle caselle annerite e tutte le lettere contenute in quelle normali; la seconda, alla quale è possibile accedere solo una volta completata la modellazione prevista dalla precedente, consente invece la gestione delle definizioni associate alle parole che compongono il cruciverba. Ora l'utente non dovrà più scegliere quali caselle numerare e rendere quindi l'origine di una parola, infatti l'applicazione le rileverà automaticamente, le elencherà in questa seconda finestra e consentirà di inserire e modificare le definizioni per ognuna di esse. La terza finestra infine rappresenterà la fine del processo e consentirà di avviare il processo di creazione dell'applicazione Flash vera e propria, oltre a mostrare del testo informativo sul cruciverba che verrà prodotto.
- Un'altra caratteristica che il programma dovrebbe presentare è l'internazionalizzazione, ovvero la possibilità scegliere la lingua nella quale visualizzare l'interfaccia dell'applicazione, con il risultato di tradurre dinamicamente in qualsiasi momento tutti i testi nel linguaggio scelto. Ciò è realizzabile eliminando ogni stringa testuale dal codice Java, come quelle visualizzate nelle etichette, nei pulsanti, ecc., ed elencandole tutte in tanti file quante sono le lingue che si vogliono supportare, in modo tale da avere ogni testo tradotto in tutti i linguaggi desiderati. Successivamente nel codice è possibile fare riferimento per ogni stringa a tali elementi contenuti nei file appena descritti, in modo tale che l'applicazione vada a ricercare dinamicamente all'interno di uno dei file il testo da stampare, a seconda della lingua attiva al momento. I linguaggi che sono stati previsti sono l'italiano e l'inglese, ma è possibile aggiungerne di supplementari.
- È utile introdurre un'opzione per consentire all'utente di salvare un cruciverba durante il processo di modellazione e quindi una per poterne caricare uno precedentemente salvato, in modo da riprendere la costruzione senza dover ricominciare dall'inizio. Queste due opzioni devono essere accessibili tramite il menù "File" presente nelle prime due finestre di modellazione del cruciverba (si è ritenuto non necessario dotare di menù l'ultima finestra in quanto rappresenta un breve passaggio conclusivo nell'intero processo di costruzione del cruciverba). L'opzione "Salva" si occuperà di creare un file contenente tutte le informazioni necessarie per poter riprendere in seguito le operazioni di generazione, mentre "Apri" darà

la possibilità di leggere uno dei file salvati tramite la precedente opzione e ricostruire il cruciverba inconcluso nell'interfaccia grafica (visualizzando la prima o la seconda finestra di modellazione a seconda dell'avanzamento del lavoro di modellazione da parte dell'utente al momento del salvataggio).

- È opportuno aggiungere al menù dell'interfaccia grafica dell'applicazione anche un'opzione chiamata proprio "Opzioni", accessibile tramite la voce di menù "Preferenze" che verrà aggiunta nelle prime due finestre di modellazione del cruciverba. In tal modo viene offerta la possibilità all'utente di scegliere ad esempio il linguaggio nel quale visualizzare l'applicazione e quale tra i file dizionario esistenti utilizzare per ottenere aiuti circa le parole da inserire nel cruciverba.

Quelle finora elencate sono le nuove caratteristiche che l'applicazione in generale deve presentare; ora invece si procede con l'analisi delle tre schermate in cui è stata suddivisa l'interfaccia grafica, osservando le funzionalità messe a disposizione dell'utente da ognuna di esse:

- **Prima schermata, modellazione della griglia del cruciverba (Fig.5):** come già detto questa prima schermata deve consentire all'utente di costruire il cruciverba definendone le dimensioni, le caselle annerite e tutte le lettere in esso contenute. Il suo aspetto grafico è basato su quello presentato dalla precedente versione dell'applicazione, in quanto la struttura della griglia del cruciverba deve rimanere pressoché invariata, ma si è deciso di porre tutti i controlli sotto ad essa piuttosto che a destra come in precedenza. La finestra viene ad essere composta da soli tre pannelli, il primo dei quali, in posizione nord, conterrà come in precedenza soltanto un'etichetta d'introduzione; il secondo invece, in posizione centrale, sarà il principale e mostrerà la griglia del cruciverba che si è deciso di estendere a 20 per 20 caselle totali (riducendo le dimensioni delle singole caselle di conseguenza); il terzo infine, in posizione sud, conterrà tutti i controlli della schermata in maniera ordinata ed elegante, presentando sei suddivisioni dalle stesse dimensioni ognuna delle quali ospiterà ed introdurrà tramite delle etichette di testo una diversa funzionalità. Elenchiamo dunque tali controlli, tra i quali ve ne sono inclusi due completamente nuovi:
 - *Settaggio delle dimensioni:* rimane pressoché invariato rispetto alla precedente versione dell'applicazione, composto da due Combo Box, uno relativo alla larghezza ed uno all'altezza del cruciverba, i quali consentono ora una scelta di valori che spazia da 3 a 20 vista la scelta di aumentare le dimensioni massime raggiungibili;

- *Cambiamento della direzione di scrittura*: anch'esso invariato rispetto alla precedente versione, composto principalmente da un pulsante con il compito di effettuare tale cambiamento e da un'etichetta di testo che indica quale dei due orientamenti è quello attivo al momento. Un'ulteriore etichetta accompagna tale controllo, con il compito di indicare la combinazione di tasti "Control" più una delle frecce direzionali in grado di svolgere la stessa funzionalità della pressione del pulsante;
- *Annerimento delle caselle*: questo controllo rappresenta la prima vera novità rispetto alla precedente versione dell'applicazione. Si tratta anche in questo caso di un pulsante con il compito di annerire la casella del cruciverba selezionata al momento, accompagnato da un'etichetta che informa della possibilità di realizzare la stessa operazione cliccando sulla casella desiderata con il tasto destro del mouse. Si è deciso di inserire nell'interfaccia grafica un pulsante anche per questa funzionalità, prima accessibile esclusivamente tramite il click con il mouse, principalmente per mantenere l'uniformità con tutti gli altri controlli che si basano su un componente grafico sul quale interagire;
- *Suggerimenti da dizionario*: segue quella che è l'ultima funzionalità implementata nella vecchia versione dell'applicazione. Come in precedenza essa è accessibile tramite un pulsante il quale si occuperà di ricostruire la parola di cui fa parte la lettera selezionata al momento e di ricercare nel dizionario le parole che si adattano negli spazi disponibili;
- *Stampa del cruciverba*: si tratta di una funzionalità completamente nuova che si è ritenuto utile aggiungere. All'interfaccia grafica viene aggiunto un ulteriore pulsante che riporta la dicitura "Stampa" e che viene introdotto da un'etichetta introduttiva che riporta il testo "Stampa cruciverba"; alla pressione del pulsante da parte dell'utente verrà lanciata una stampa del pannello dell'interfaccia contenente la griglia del cruciverba, in modo da consentire, qualora se ne avesse il bisogno, di averne un riscontro visivo sempre disponibile, anche ad esempio nel momento in cui si passa alla schermata successiva dell'applicazione, dove tale struttura non sarà più visualizzata;
- *Controlli principali*: nella sesta e ultima suddivisione del pannello dei controlli della prima schermata dell'interfaccia grafica dell'applicazione sono presenti i due pulsanti principali per la navigazione nel processo di costruzione del cruciverba,

ovvero i pulsanti “Esci” per la chiusura del programma e “Continua” per proseguire alla seconda finestra.

Questa prima schermata dell’interfaccia grafica è inoltre dotata di un menù che presenta le voci che già si erano incontrate nella vecchia versione dell’applicazione (l’opzione per la chiusura del programma, quella per ottenere informazioni sull’applicazione stessa e quella per visualizzarne le istruzioni per l’uso) alle quali si aggiungono ora quelle citate in precedenza, ovvero le opzioni per salvare il cruciverba che si sta costruendo ed aprirne uno salvato in precedenza e quella per poter impostare le preferenze in aspetti quali il linguaggio usato dall’applicazione ed il file da cui leggere il dizionario desiderato. Ulteriori modifiche da citare rispetto alla vecchia versione dell’interfaccia grafica sono il colore dello sfondo della casella selezionata che da giallo passa ad azzurro (nella precedente applicazione tale colore era utilizzato per identificare le caselle numerate, ora non più necessario) e un più efficiente controllo sui caratteri digitati nelle caselle del cruciverba, consentendo ora l’inserimento delle sole lettere dell’alfabeto ed escludendo ogni altro simbolo.

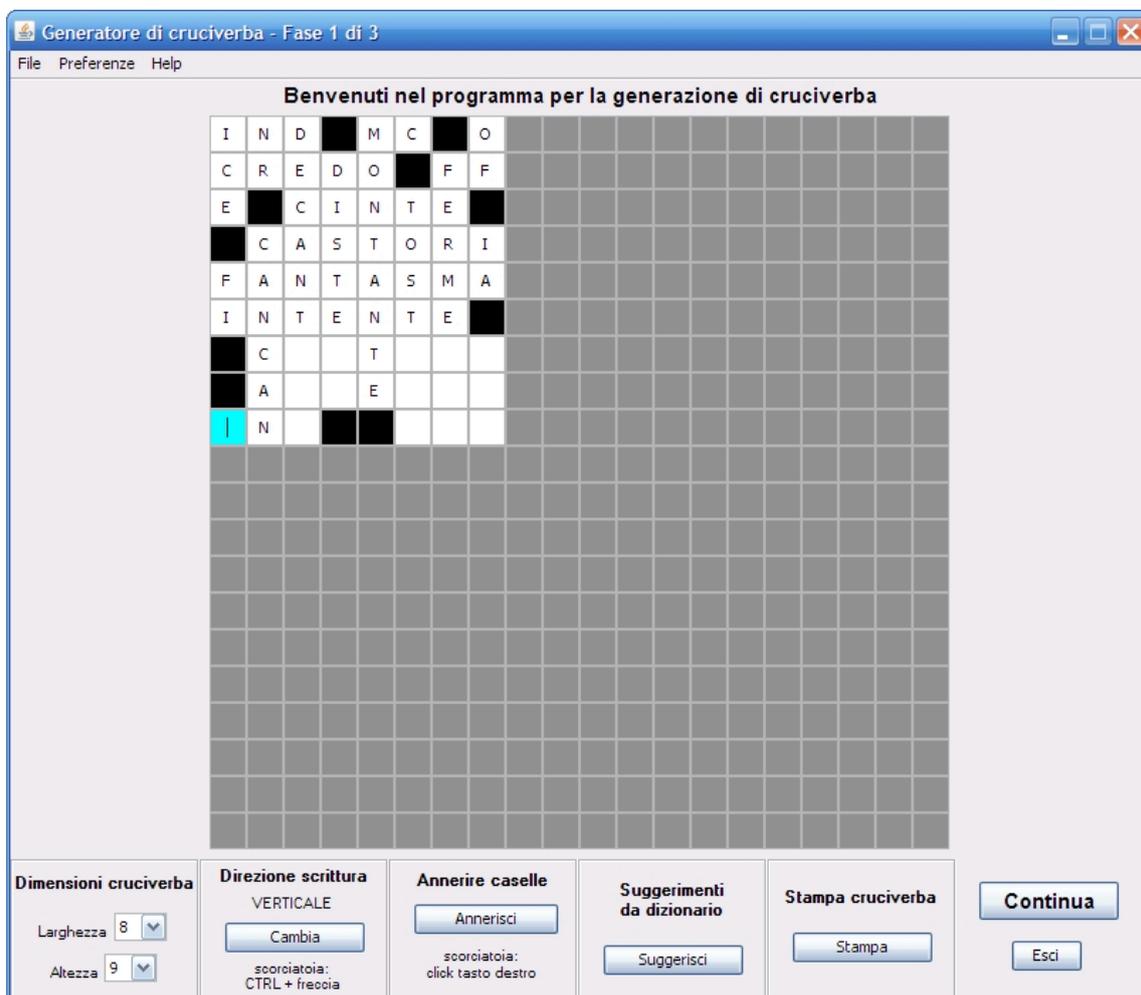


Fig.5: Screenshot della prima schermata dell’interfaccia grafica del generatore di cruciverba

– **Seconda schermata, gestione delle definizioni (Fig.6):** questa seconda schermata è completamente nuova rispetto alla precedente versione dell'applicazione e, come detto in precedenza, deve mostrare le parole che compongono il cruciverba così come è stato costruito nella precedente schermata e consentire all'utente di associare ad ognuna di esse una definizione (nonché di rimuoverle e di modificarle). La sua struttura a pannelli non si discosta completamente da quella presentata dalla prima finestra, in quanto anch'essa ne presenta tre di principali con la stessa disposizione di quelli che compongono la schermata precedente. Di seguito è descritto il loro contenuto:

- *Pannello nord (introduzione):* come nella precedente schermata, questo pannello contiene esclusivamente un'etichetta introduttiva alla finestra stessa;
- *Pannello centrale (liste delle parole):* quello centrale è il pannello principale ed in esso vengono visualizzate due liste contenenti le parole che compongono il cruciverba, una per quelle alle quali non è ancora stata assegnata una definizione (inizialmente quindi si troveranno tutte in questa) ed una per quelle alle quali è già stata inserita una definizione. La prima delle due liste si trova nella parte sinistra del pannello, mentre la seconda copre la parte destra ed è più larga della precedente in quanto dovrà visualizzare non soltanto un elenco di parole, ma anche la loro definizione accanto ad ognuna di esse, quindi ogni riga conterrà un testo molto più lungo;
- *Pannello sud (controlli):* anche in questa schermata, come nella precedente, tutti i controlli sono posti al di sotto del resto dei componenti grafici. Questo pannello viene ulteriormente diviso in due parti, una a nord ed una a sud, la prima delle quali ha il compito di consentire l'inserimento e la modifica della definizione relativa alla parola selezionata da una delle liste soprastanti presentando un campo di testo in cui essa può essere digitata, un pulsante riportante la dicitura "Salva definizione" per memorizzare il testo digitato nel campo di testo ed un'etichetta che riporta le coordinate della parola alla quale si sta assegnando la definizione (ad esempio "1 orizzontale"). La parte inferiore del pannello invece è suddivisa in tre parti nello stesso modo in cui quello della schermata precedente era diviso in sei, in modo tale da mantenere lo stesso stile, ognuna delle quali contiene il controllo per attivare una particolare funzionalità.

Passiamo ora in rassegna tutte le funzionalità messe a disposizione dall'applicazione in questa schermata:

- *Selezione di una parola dalle due liste:* come già detto nel pannello centrale della schermata sono visualizzate le due liste contenenti tutte le parole che compongono il cruciverba suddivise tra quella a sinistra (prive di definizione) e quella a destra (con una definizione assegnata). L'utente ha la possibilità di selezionare una qualsiasi parola da una delle due liste in qualsiasi momento in modo tale da vederne apparire le coordinate nell'etichetta del pannello dei controlli, accanto al campo di testo per l'inserimento della definizione, mentre proprio in quest'ultimo verrà mostrata la definizione associata alla parola selezionata (se ve n'è una, ovvero se tale parola si trova nella lista di destra);
- *Inserimento e salvataggio delle definizioni:* quando viene selezionata una parola da una delle due liste, come detto qui sopra, ne viene visualizzata la definizione associata nel campo di testo apposito (sempre che ve ne sia una da riportare), mentre nella vicina etichetta vengono mostrate le sue coordinate. A questo punto è possibile per l'utente digitare la definizione desiderata nel campo di testo o di modificare o cancellare quella già presente, dopodiché, per registrare tali operazioni e quindi salvare la definizione presente al momento nel campo di testo, è sufficiente premere il pulsante "Salva definizione" adiacente, il quale si occupa di aggiornare il database con la nuova informazione;
- *Visualizzare il cruciverba:* questo è il primo dei tre controlli presenti nella parte inferiore del pannello di controllo dell'interfaccia grafica e presenta una funzionalità che è stata inserita per fornire la possibilità di osservare la struttura del cruciverba modellato nella schermata precedente mentre si opera in questa seconda qualora sia necessario. Si tratta semplicemente di un pulsante alla pressione del quale verrà visualizzata una finestra contenente la griglia del cruciverba con tutte le lettere inserite (un po' come vederlo nell'applicazione Flash finale con attivata l'opzione per mostrarne la soluzione). Tale finestra non fornisce ulteriori funzionalità, ma esclusivamente un pulsante per chiuderla quando non si ha più bisogno di consultarla;
- *Suggerimenti da un dizionario online:* questa nuova funzionalità, la seconda tra quelle presenti nella parte inferiore del pannello di controllo, si occupa di cercare informazioni riguardanti la parola selezionata al momento su un dizionario online. Si tratta di un pulsante alla pressione del quale viene aperta automaticamente la pagina web relativa alla parola interessata tra quelle che compongono il dizionario online nel browser predefinito del sistema in uso, con lo scopo di avere delle tracce sulla

definizione da inserire o addirittura delle definizioni vere e proprie che è possibile copiare. Per svolgere queste operazioni l'applicazione supporta tre versioni del dizionario Wikizionario, progetto parallelo a quello di Wikipedia, ed in particolare le versioni italiana, inglese e tedesca, ma potrebbe essere esteso con l'inserimento di ulteriori dizionari;

- *Controlli generali:* come per la precedente schermata, l'ultima suddivisione della parte inferiore del pannello di controllo dell'interfaccia grafica ospita i controlli necessari per la navigazione all'interno dell'applicazione: anche in questo caso sono messi a disposizione due pulsanti, uno per la chiusura del programma e l'altro per accedere alla terza ed ultima schermata una volta terminato l'inserimento di tutte le definizioni del cruciverba.

Infine è necessario ricordare che anche questa seconda schermata è dotata di un menù che mette a disposizione le opzioni già viste in quello della finestra precedente, con i controlli quindi per il salvataggio ed il caricamento di cruciverba in fase di modellazione, per l'uscita dall'applicazione, per il settaggio delle preferenze (tra cui, oltre a quelle già citate in precedenza, da aggiungere anche quella che permette la scelta di quale tra i dizionari online supportati utilizzare per ottenere suggerimenti), per la visualizzazione delle istruzioni per l'uso e delle informazioni sull'applicazione.

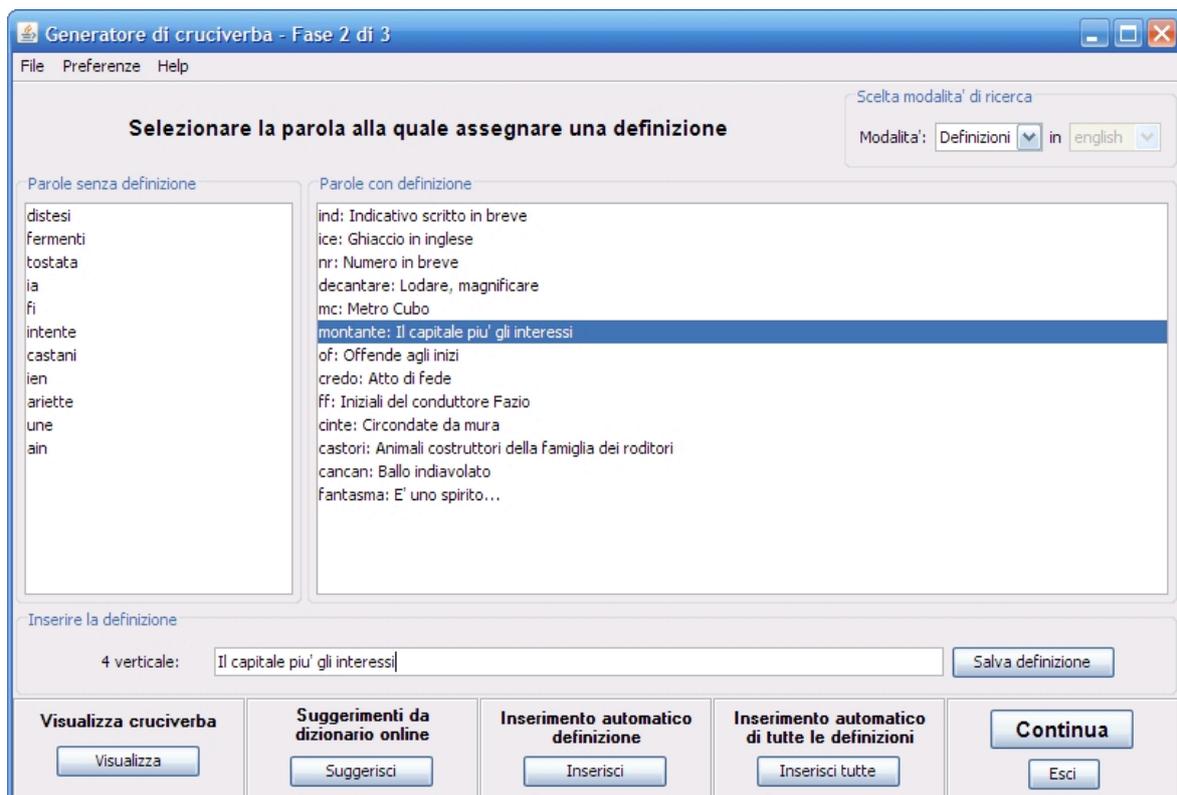


Fig.6: Screenshot della seconda schermata dell'interfaccia grafica del generatore di cruciverba

– **Terza schermata, conclusione (Fig.7):** quest’ultima schermata presentata dall’applicazione conclude l’intero processo di costruzione del cruciverba e, come le precedenti, è strutturata su tre pannelli nelle posizioni nord, sud e centrale. L’unica operazione richiesta all’utente prima della pressione del pulsante “Genera cruciverba” è l’inserimento del nome da assegnare ad esso da digitare in un apposito campo di testo. Di seguito sono elencati e descritti i pannelli e la loro struttura:

- *Pannello nord (introduzione):* come già nelle due precedenti finestre, anche in questa l’unico componente presente nel pannello nord è un’etichetta contenente del testo che introduce l’utente alla schermata;
- *Pannello centrale (corpo principale):* questo è il pannello principale della schermata ed è ulteriormente suddiviso in due parti, una in posizione nord ed una più grande centrale. La parte superiore contiene il campo di testo in cui l’utente può digitare il nome scelto per il cruciverba costruito ed un’etichetta che ne indica la natura, mentre quella sottostante è rappresentata da un’area di testo in cui vengono mostrate alcune indicazioni riguardanti l’applicazione Flash che si sta generando: in particolare si rende noto che il nome scelto in questa schermata sarà assegnato alla cartella che si sta per costruire nella quale verrà creato il cruciverba e che quindi tale cartella può essere spostata e copiata ovunque a piacimento visto che una volta terminato il processo essa risulterà totalmente indipendente dall’applicazione Java generatrice. Infine si comunica che per avviare il cruciverba in Flash è sufficiente aprire la pagina “index.html” che si troverà nella cartella creata;
- *Pannello sud (controlli):* i controlli principali, come nelle schermate precedenti, si trovano nella parte bassa della finestra e consistono in due pulsanti, ovvero quello per chiudere l’applicazione e quello per avviare il vero e proprio processo di generazione dell’applicazione Flash.

Come già accennato in precedenza questa finestra non è dotata di menù come le precedenti in quanto essa costituisce soltanto una breve parentesi conclusiva nella costruzione del cruciverba ed inoltre molte delle funzionalità accessibili dai menù presentati per le precedenti schermate non sarebbero utili in questa.

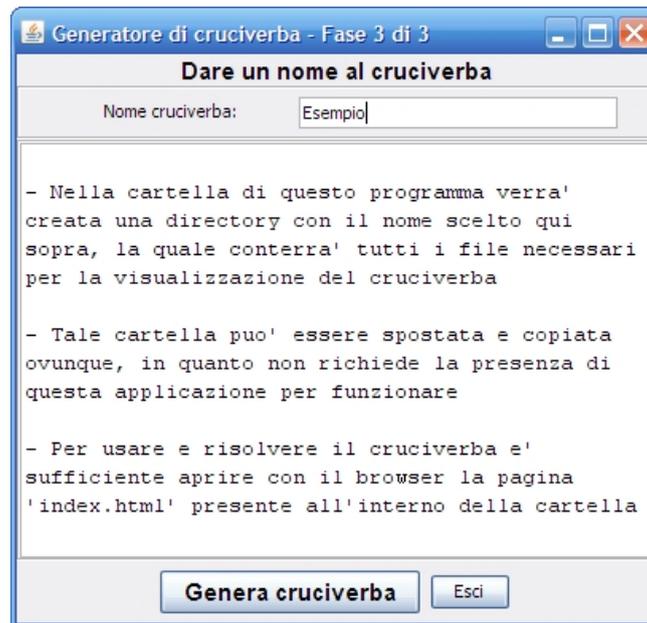


Fig.7: Screenshot della terza schermata dell'interfaccia grafica del generatore di cruciverba

Passiamo ora ad analizzare questa nuova versione dell'applicazione da un punto di vista maggiormente tecnico basandoci, come fatto per la versione precedente, sull'interazione che può essere svolta dall'utente.

Va da dire innanzitutto che l'organizzazione del codice è fondamentalmente rimasta invariata rispetto a prima, infatti viene utilizzata nuovamente l'architettura *three-tier* e le classi esistenti rimangono tali, seppur con delle modifiche, alle quali però se ne aggiungono di nuove.

Tra queste ultime va citata subito quella denominata *Word*, la quale ha un compito molto simile a *Cell*, già vista nella precedente versione dell'applicazione, ovvero costituire un'unità base di un database, ed in particolare quello che sarà ampiamente utilizzato nell'ambito della seconda schermata dell'interfaccia grafica, ovvero nel momento in cui si deve lavorare sulle parole che compongono il cruciverba e la loro definizione. In seguito sono elencati gli attributi che descrivono un oggetto derivante da tale classe:

- stringa *word*: questa è la stringa contenente la parola vera e propria descritta dall'oggetto in questione;
- stringa *def*: contiene la definizione associata alla parola a cui fa riferimento l'oggetto;
- variabile booleana *dir*: sta ad indicare il senso seguito dalla parola all'interno del cruciverba con il valore "false" in caso di orientamento orizzontale, "true" invece in caso di orientamento verticale;

- intero *num*: è il numero della definizione associata alla parola e quindi della casella che dà origine ad essa, che insieme a *dir* individua univocamente la posizione della parola stessa all'interno del cruciverba.

Ora ha inizio l'elenco delle funzionalità messe a disposizione dall'applicazione a partire dalle quali è possibile compiere un'analisi completa dei processi nascosti all'utente che vengono svolti per rendere il programma in grado di soddisfare i requisiti per cui è stato implementato.

- **Inizializzazione**: per prima cosa è bene osservare tutte le operazioni svolte all'avvio dell'applicazione prima che venga visualizzata l'interfaccia grafica e di conseguenza prima che inizi il dialogo con l'utente, poiché ve ne sono numerose in più rispetto alla vecchia versione. Ciò che rimane invariato rispetto a prima è l'iniziale creazione di un'istanza della classe *Manager*, la quale successivamente si occupa di tutte le operazioni preliminari necessarie per l'attivazione di tutte le funzionalità messe a disposizione dall'applicazione: per primo viene creato un vettore nel quale vengono inseriti degli oggetti derivanti dalla classe Java *Locale*, ognuno dei quali rappresenta una particolare regione geografica (con il suo linguaggio, la sua cultura, ecc.) ed in particolare quelle per le quali si desidera avere la possibilità di tradurre nella lingua adeguata ogni singolo testo dell'interfaccia grafica, realizzando così la proprietà di internazionalizzazione. Durante lo sviluppo sono state previste le lingue italiana ed inglese, poiché questi due sono i *Locale* inseriti nel vettore appena citato, ma è sempre possibile estendere tale supporto ad altri linguaggi. Subito dopo uno di questi *Locale* viene selezionato come attivo (si tratta di quello predefinito, titolo assegnato a quello corrispondente alla lingua italiana) inserendolo in un'apposita variabile con lo scopo di identificare quello che detiene tale proprietà. L'operazione successiva è quella che vede la creazione di un'istanza della classe *Dictionary* proprio come avveniva anche nella versione precedente dell'applicazione, ed anche in questo caso, nella fase della sua costruzione, essa si occupa di inizializzare un vettore contenente tutte le parole lette da un file dizionario (di predefinito anche in questo caso viene letto quello in lingua italiana fornito con l'applicazione, dopodiché è possibile cambiare dizionario durante l'utilizzo dell'applicazione). Diversamente dalla versione precedente però ora *Manager* richiede subito un servizio reso disponibile da *Dictionary*: la richiesta è quella di costruire un vettore nel quale inserire le informazioni sui dizionari online supportati dall'applicazione (come detto in precedenza si tratta delle versioni italiana, inglese e tedesca del Wikizionario alle quali è comunque possibile aggiungerne altre, così come anche diversi dizionari). I dati

necessari vengono raccolti da un file testuale fornito con l'applicazione (ogni riga contiene i dati necessari ad un dizionario online), quindi viene creata un'istanza della nuova classe *OnlineDictionary* per ogni dizionario descritto in tale file, ed ognuna di esse viene ad essere un elemento del vettore creato da *Dictionary*. La classe *OnlineDictionary* ha il solo scopo di contenere tutti i dati riguardanti ogni singolo dizionario, così come già accadeva per le caselle del cruciverba con la classe *Cell* nella precedente versione dell'applicazione, ed anch'essa verrà a far parte del *data tier* nell'architettura *three-tier* dell'applicazione. Di seguito sono elencati gli attributi da essa presentati:

- Stringa *name*: contiene il nome del dizionario, così come verrà presentato nell'applicazione;
- Stringa *link*: contiene l'indirizzo web di base tramite il quale è possibile accedere al dizionario, dal quale è possibile costruire gli indirizzi specifici per le singole parole sulle quali si vogliono avere informazioni;
- Stringa *lang*: indica la lingua presentata dal dizionario.

Una volta riempito il vettore con i dati sopra elencati per ogni dizionario online disponibile, il controllo torna a *Manager*, il quale si occupa di estrarne il primo elencato e memorizzarlo in un'apposita variabile che lo identifica come attualmente in uso (di predefinito sarà il Wikizionario il lingua italiana). Infine, le ultime operazioni svolte da *Manager* in fase di inizializzazione riguardano la costruzione delle tre schermate in cui è suddivisa l'interfaccia grafica dell'applicazione, la quale si risolve nella creazione di un'istanza di ognuna delle tre classi che le costituiscono, ovvero *Crossword* per la prima, *Definitions* per la seconda ed *End* per la terza, le tre principali che costituiscono il *presentation tier* nel complesso dell'architettura *three-tier* sulla quale è basata l'applicazione. Durante la loro costruzione si presta attenzione ad una proprietà citata in precedenza, ovvero l'internazionalizzazione dell'applicazione: come già detto sono stati approntati due file testuali contenenti tutte le stringhe da visualizzare nell'interfaccia grafica nelle due lingue supportate, italiano ed inglese, ed ora le tre finestre devono andare a prelevare quelle di cui hanno bisogno da uno dei due file, ovvero quello della lingua attiva di predefinito, l'italiano. Per fare ciò si ricorre alla classe Java *ResourceBundle*, un oggetto della derivante dalla quale è in grado proprio di fare riferimento a delle risorse esterne al codice delle quali quest'ultimo si può servire, in questo caso uno dei file contenenti tutte le stringhe nel linguaggio attivo. Tale istanza di *ResourceBundle* viene quindi creata in modo tale da avere un riferimento al file relativo al

Locale attivo al momento, quindi di predefinito quello italiano. Avendo in questo modo le risorse a disposizione, ognuna delle tre finestre in questa sua fase di costruzione si occupa di eseguire un metodo con il compito di stampare tutte le stringhe di testo che dovranno essere presentate da tale schermata andandole a recuperare nel file relativo al linguaggio attivo di predefinito, ovvero l'italiano, tramite una chiave ad esse assegnata, chiaramente uguale per ognuno dei linguaggi disponibili. A questo punto è possibile rendere disponibile l'interfaccia grafica all'utente in modo da consentirne l'interazione, quindi la prima delle tre finestre appena costruite viene visualizzata sullo schermo. È da notare una differenza rispetto alla precedente versione dell'applicazione: in precedenza in questa fase di inizializzazione veniva già costruito anche il vettore in cui si memorizzano i dati di tutte le caselle che compongono il cruciverba, e tali dati venivano in esso inseriti durante la fase di modellazione del cruciverba da parte dell'utente. Ora invece tale vettore verrà costruito solo in un secondo momento ed in particolare solo alla fine della prima parte della lavorazione dell'utente, cioè nel momento in cui viene ultimata la modellazione del cruciverba nella prima schermata dell'interfaccia grafica e si sceglie di proseguire con la seconda.

– **Funzionalità della prima schermata:**

- *Annerire una casella:* questa operazione può essere eseguita in due modi ottenendo lo stesso risultato, ovvero premere il pulsante apposito tra i controlli dell'interfaccia grafica dopo aver selezionato la casella che si vuole annerire o cliccare con il tasto destro del mouse su di essa; i metodi che vengono eseguiti al presentarsi di queste due interazioni dell'utente si occupano di controllare lo stato attuale della casella interessata, osservandone il colore di sfondo e verificando se essa è abilitata o meno, venendo quindi a sapere se essa è attualmente già annerita o meno. A seconda di ciò i due metodi svolgono le operazioni necessarie: nel caso in cui la casella sia normale lo sfondo viene colorato di nero, viene rimosso l'eventuale testo in essa contenuto e viene disabilitata; in caso contrario il colore di sfondo torna ad essere bianco e la casella viene riabilitata. A differenza della precedente versione dell'applicazione non viene richiesto a *Manager* di registrare queste modifiche, infatti il database (ovvero il vettore contenente i dati di ogni casella del cruciverba) non è ancora stato creato; ciò avverrà, come già detto, solo alla fine della modellazione eseguita in questa prima schermata dell'interfaccia grafica;
- *Selezionare una casella:* il metodo eseguito all'esecuzione di questa operazione da parte dell'utente è estremamente più semplice rispetto a quello presentato dalla vecchia versione dell'applicazione: ora non è più necessario infatti recuperare dati

riguardanti tale casella, come ad esempio se vi sono delle definizioni ad essa associata o se essa è stata scelta come numerata o meno, è infatti sufficiente registrare la casella selezionata in un'apposita variabile, in modo che ogni metodo sia a conoscenza di tale informazione, e colorarne lo sfondo in azzurro, ovvero il colore scelto per indicare la casella selezionata al momento;

- *Digitare lettere nelle caselle, spostarsi tra di esse e cambiare direzione di scrittura:* queste operazioni, tutte eseguibili tramite la pressione di determinati tasti mentre una casella del cruciverba è selezionata esattamente come avveniva nella precedente versione dell'applicazione, vengono realizzate tramite un metodo attivato nel momento in cui l'utente svolge tale interazione. Esso opera in maniera assolutamente identica al suo predecessore, quindi per prima cosa si occupa di controllare che non sia stata premuta la combinazione di tasti "Control" più una delle frecce direzionali, nel qual caso deve essere cambiata la direzione di scrittura, per cui vengono modificati il valore della variabile booleana che la indica ed il testo contenuto nell'etichetta che la riporta, quest'ultimo recuperato da uno dei file risorsa contenenti tutte le stringhe testuali dell'interfaccia grafica in uno dei linguaggi supportati grazie a *ResourceBundle*. Se il tasto premuto è un singolo la precedente versione di questo metodo ne riportava il carattere relativo (se il tasto in questione ne ha uno associato) senza verificare che si trattasse di una lettera come in effetti dovrebbe essere; ora invece al contrario viene inserita il carattere nella casella selezionata rimuovendo qualsiasi cosa vi fosse contenuta precedentemente solo nel caso si tratti proprio di una lettera, mentre in caso contrario non si riscontrerà alcun effetto evidente poiché un eventuale carattere diverso (come ad esempio quelli di punteggiatura) digitato verrà immediatamente rimosso dal metodo in analisi. Infine, come avveniva nella precedente versione dell'applicazione, viene gestito lo spostamento da effettuare alla pressione del tasto, ed in particolare se viene premuta una delle frecce direzionali verrà selezionata la casella adiacente a quella di partenza nella direzione indicata dalla freccia stessa (sempre che ciò sia possibile, ovvero se non si è raggiunto un bordo del cruciverba o se la casella in cui si cerca di spostarsi è annerita e quindi disabilitata), mentre se viene premuta una lettera, e non un altro tasto qualsiasi come in precedenza, dopo aver stampato il carattere nella casella selezionata, verrà automaticamente effettuato lo spostamento nella casella adiacente nella direzione indicata dalla variabile booleana che la indica, quindi verso destra in caso di

direzione orizzontale e verso il basso in caso di direzione verticale, anche in questo caso solo se tale spostamento è consentito dalla morfologia del cruciverba.

- *Cambiare le dimensioni del cruciverba*: anche il metodo eseguito associato a questa operazione, il quale viene eseguito nel momento in cui l'utente interagisce con i due Combo Box che rappresentano altezza e larghezza del cruciverba, risulta molto più semplice di quello visto nella vecchia versione dell'applicazione, infatti anche in questo caso non avviene alcuna comunicazione con il *Manager* e quindi con il database. In questa riprogettazione esso si limita a raccogliere le due misure impostate dall'utente tramite i Combo Box ed a confrontarle con le coordinate di tutte le quattrocento (20 per 20) caselle di cui è composta la griglia grafica svolgendo le seguenti operazioni: se una casella presenta almeno una delle due coordinate che supera le dimensioni impostate questa viene immediatamente disabilitata ed il suo sfondo viene colorato in grigio; se invece si trova all'interno dei margini impostati dall'utente viene verificato il suo stato e, nel caso si rilevi che essa era in precedenza disabilitata e che presentava lo sfondo grigio (si trovava quindi oltre le dimensioni scelte), essa viene riabilitata ed il suo sfondo torna ad essere bianco. I cambiamenti apportati da questo metodo sono anche in questo caso esclusivamente grafici e non vengono registrati in alcun database;
- *Cambiare la direzione di scrittura*: il metodo eseguito alla pressione del pulsante apposito da parte dell'utente non svolge nulla di nuovo rispetto a quanto già visto per quello associato alla battitura di un tasto mentre una delle caselle è selezionata. In particolare sono qui previste le stesse operazioni svolte alla pressione della combinazione di tasti "Control" più una delle frecce direzionali, ovvero il valore della variabile booleana che sta ad indicare la direzione di scrittura attualmente attiva viene invertito, ed anche il testo contenuto nell'etichetta che accompagna il pulsante a cui si sta facendo riferimento viene modificato in maniera tale da visualizzare quello corretto tra i due orientamenti, orizzontale oppure verticale;
- *Ottenere suggerimenti da dizionario*: gran parte del metodo attivato alla pressione del pulsante con il compito di svolgere questa operazione opera nello stesso modo di quello presente nella precedente versione dell'applicazione. Come esso infatti rileva la casella selezionata al momento dell'attivazione di questa funzionalità, scorre a ritroso le caselle (verso sinistra o verso l'alto a seconda della direzione di scrittura attiva) fino ad individuare quella da cui la parola ha origine, ricostruisce la parola inserendo carattere per carattere in una stringa, così come è presentata dalle caselle

del cruciverba (sostituendo con un asterisco ‘*’ le lettere mancanti), ne verifica l’incompletezza (se al contrario fosse già completa lo comunicherebbe all’utente tramite un *Dialog* interrompendo il processo) ed invia una richiesta alla classe *Dictionary* per la ricerca nel dizionario attivo delle parole che si adattano agli spazi a disposizione per poi stamparle in una finestra che le elenca. L’unica differenza che diversifica le due versioni di questo metodo è l’inserimento di un nuovo controllo che in questa riprogettazione dell’applicazione regola la ricerca la filtrazione di tutte le voci contenute nel dizionario in uso: il metodo messo a disposizione da *Dictionary* infatti non sfoglierà sempre tutte le voci presenti nel dizionario fino alla fine come avveniva in precedenza; ora interromperà il filtraggio non appena verranno isolate le prime mille parole adatte, ciò perché in caso di dizionari ben forniti e di parole lunghe e con molte lettere mancanti il processo potrebbe rilevare un numero molto grande di parole adatte durante la ricerca, tanto da caricare di lavoro in maniera eccessiva la memoria del calcolatore e da richiedere un tempo di esecuzione che può dilungarsi in modo fastidioso e poco utile. L’utente potrebbe però non essere soddisfatto da una risposta parziale che conta solo mille parole quando in realtà ve ne sarebbero molte di più di valide, per cui il metodo in analisi procederà con la visualizzazione di un *Dialog* in grado di notificare l’accaduto all’utente e di consentirgli due scelte: accettare la ricerca interrotta e consultarne immediatamente il risultato oppure forzare una ricerca completa senza più interruzioni. È chiaro che nel primo caso verrà subito mostrata la finestra con l’elenco dei mille vocaboli ed il processo sarà ultimato, ma nel caso in cui l’utente richieda una ricerca completa verrà inoltrata una seconda richiesta di filtraggio a *Dictionary*, questa volta specificando di ignorare il controllo sulla quantità di risultati utili rilevati, la quale verrà gestita così come avveniva nella precedente versione dell’applicazione, concludendo anche in questo caso con la visualizzazione di tutti i vocaboli filtrati nella finestra generata da un’istanza della classe *Suggest*;

- *Stampare il cruciverba*: questa nuova funzionalità offerta dalla nuova versione dell’applicazione è attivabile premendo il pulsante apposito tra quelli dell’interfaccia di controllo, interazione che causa l’esecuzione del metodo con il compito di svolgere tutte le operazioni necessarie per realizzare la stampa. La prima cosa da fare è preparare la griglia del cruciverba rimuovendone i dettagli che non si vuole vengano mostrati nella stampa, ed in particolare la colorazione azzurra della casella attualmente selezionata, la quale viene individuata e colorata di bianco.

Successivamente il metodo si occupa di creare un'istanza di una nuova classe di supporto denominata *PrintableComponent*, inserita nel *logic tier* dell'architettura *three-tier* e contenente tutti gli strumenti necessari per portare a termine l'intero processo di stampa, e di avviare tale processo inoltrando una richiesta all'oggetto appena creato comunicandogli il pannello dell'interfaccia grafica corrispondente alla griglia del cruciverba, ovvero ciò che dovrà essere stampato. *PrintableComponent* utilizza la classe Java principale per il controllo della stampa, ovvero *PrinterJob*, per svolgere tutte le operazioni necessarie per approntare la stampa del pannello del cruciverba comunicato dal metodo in esame, visualizzando tra l'altro un *Dialog* di configurazione di stampa tramite il quale l'utente può impostare le proprie preferenze.

- *Visualizzare le informazioni riguardanti l'applicazione*: come avveniva nella precedente applicazione queste informazioni sono accessibili tramite la voce "About" nel menù "Help" presentato da questa schermata dell'interfaccia grafica e vengono visualizzate in una finestra che viene mostrata appositamente e che presenta un unico controllo con il quale l'utente può interagire, ovvero un pulsante per la sua chiusura. La finestra in questione viene realizzata istanziando un oggetto dalla classe *About*, contenuta nel *presentation tier* dell'architettura;
- *Visualizzare le istruzioni per l'uso delle funzionalità della prima schermata*: anche queste istruzioni vengono mostrate tramite la visualizzazione di una finestra molto simile a quella della funzionalità precedente. Questo comando è raggiungibile tramite la voce "Help" nell'omonimo menù di questa prima schermata dell'interfaccia grafica e la classe da cui l'istanza della finestra delle istruzioni viene ottenuta è *CrosswordHelp*, anch'essa parte del *presentation tier* dell'architettura;
- *Impostare le proprie preferenze*: questa funzionalità è accessibile tramite la voce "Opzioni" nel menù "Preferenze" e nel momento in cui l'utente interagisce con essa viene eseguito un metodo con il compito di creare un'istanza della nuova classe *Options*, la quale costituisce una finestra dalla quale è possibile impostare le preferenze che l'applicazione deve rispettare, e quindi di visualizzarla. Tale finestra si compone di quattro parti disposte orizzontalmente di cui le prime tre fanno riferimento ad un'opzione differente che è possibile impostare a piacimento tramite un Combo Box che rende disponibili le diverse alternative, mentre l'ultima contiene i pulsanti per annullare l'operazione tornando alla schermata del cruciverba e per confermare le modifiche operate. La prima sezione riguarda il linguaggio adottato

dall'applicazione stessa, cioè in tutte le stringhe testuali mostrate dall'interfaccia grafica. Le alternative messe a disposizione dal Combo Box vengono rilevate in fase di costruzione della finestra, quando essa analizza i *Locale* inizializzati dall'applicazione con un file delle risorse (in cui si trovano tutte le stringhe tradotte nel relativo linguaggio) associato ad ognuno essi, quindi di predefinito si tratta di italiano ed inglese; la finestra si fa inoltre comunicare da *Manager* quale di essi è il *Locale* attivo al momento, così da visualizzarlo come già selezionato nel Combo Box. La seconda sezione della finestra permette di scegliere il file dizionario da cui ottenere dei suggerimenti per quanto riguarda le parole che si adattano nella posizione scelta del cruciverba; in fase di costruzione della finestra, essa va a rilevare tutti i file presenti nella directory "dict" che viene fornita insieme all'applicazione e nella quale vanno inseriti i file di questo tipo (di predefinito sono già presenti un ben fornito dizionario italiano ed alcuni molto basilari in altri linguaggi), inserendo il nome di ogni file trovato nel Combo Box tramite il quale selezionare quello desiderato. Come accadeva per i *Locale* nella sezione precedente, anche in questo caso viene impostato come già selezionato il dizionario attivo al momento, informazione che viene richiesta ed ottenuta da *Manager*. La terza sezione di questa finestra delle preferenze consente la scelta del dizionario online tra quelli supportati da utilizzare per ricercare informazioni e definizioni sulle parole che compongono il cruciverba (funzionalità che sarà descritta meglio più avanti). Come detto in precedenza, in fase di inizializzazione dell'applicazione *Dictionary* si occupa, su richiesta di *Manager*, di leggere da un apposito file tutti i dizionari online supportati dall'applicazione (di predefinito il Wikizionario nelle sue versioni italiana, inglese e tedesca) e di descrivere ognuno di essi tramite un'istanza della classe *OnlineDictionary*, inserendola poi in un vettore. La finestra, in fase di costruzione, legge tale vettore e riporta nel Combo Box appropriato tutti i diversi dizionari rilevati, ponendo come selezionato quello attualmente in uso. La quarta sezione di questa finestra contiene due pulsanti, quello per annullare le modifiche e tornare alla modellazione del cruciverba e quello per confermare le impostazioni selezionate. Nel momento in cui l'utente preme il pulsante per dare conferma delle preferenze scelte, viene avviato un processo che opera tutti i cambiamenti che ne conseguono: per prima cosa viene richiesto a *Manager* di impostare il *Locale* selezionato come attivo, che a sua volta lo comunica alle tre classi che rappresentano le tre schermate dell'interfaccia grafica, forzandole a ricostruire il loro oggetto *ResourceBundle*, che

si riferirà ora al file del nuovo linguaggio scelto, ed a rieseguire il metodo che si occupa di stampare tutte le stringhe di testo all'interno dell'interfaccia stessa, in modo da rendere graficamente effettivo il cambiamento di linguaggio. Lo stesso avviene per il nuovo dizionario online scelto: esso viene indicato a *Manager* che lo imposta come attivo, comunicandolo a sua volta alle classi interessate che da quel momento lo utilizzeranno come attivo. Per quanto riguarda infine il file dizionario dal quale ottenere le parole che si adattano nella posizione scelta nel cruciverba, il nome di quello scelto tramite il Combo Box viene comunicato anche in questo caso a *Manager*, il quale si occupa di richiedere alla classe *Dictionary* il caricamento delle parole dal nuovo file scelto. Come avveniva in fase di inizializzazione quindi *Dictionary* apre il nuovo file e memorizza in un vettore tutti i vocaboli in esso contenuti perdendo il contenuto di quello precedente. E con ciò ha termine il processo di impostazione delle preferenze; la finestra delle opzioni viene chiusa e si torna alla lavorazione del cruciverba;

- *Salvare in un file il dizionario in lavorazione*: questa nuova funzionalità è accessibile tramite la voce “Salva” nel menù “File” presentato dalla schermata dell'interfaccia grafica e, nel momento in cui viene attivata dall'utente, svolge una serie di operazioni che hanno come risultato la memorizzazione in un file di tutte le informazioni riguardanti il cruciverba in costruzione già immesse dall'utente, in modo tale da poter riprendere il lavoro interrotto in futuro. Per prima cosa il metodo eseguito in seguito all'interazione dell'utente con il comando “Salva” si occupa di visualizzare un *Dialog* predefinito tramite il quale è possibile digitare il nome del file in cui si vuole salvare il cruciverba e scegliere il percorso in cui crearlo tramite l'esplorazione di cartelle e file del calcolatore in uso (viene suggerito un percorso predefinito, ovvero una cartella chiamata “save” che è parte integrante dell'applicazione). L'utente può scegliere di annullare il processo oppure di confermare il nome del file scelto, ed in questo secondo caso il metodo si occuperà di verificare che non esista già un file con lo stesso nome di quello inserito dall'utente; se ne viene rilevato uno verrà mostrato un altro *Dialog* che consentirà di sovrascrivere il contenuto di quello esistente o di tornare alla scelta del nome per sceglierne uno differente. Una volta confermate le informazioni per il file, il processo prosegue con la raccolta di tutti i dati immessi dall'utente, ovvero le dimensioni attuali del cruciverba e lo stato di ogni casella (se essa è annerita o meno e l'eventuale lettera digitata in essa). Viene quindi costruito un vettore contenente

una stringa per ogni casella, le quali a loro volta elencano tutti i dati necessari per descrivere la casella alla quale si riferiscono, cioè in pratica vengono elencati in breve gli stessi attributi già visti nella classe *Cell*. Infatti ogni stringa contiene le seguenti informazioni suddivise da un carattere separatore: l'identificatore della casella stessa (che include quindi anche le sue coordinate), la lettera in essa contenuta (oppure uno spazio se essa risulta ancora vuota o un asterisco "*" se essa è annerita) ed infine le tre variabili numeriche che stanno ad indicare le definizioni associate alla casella, ovvero i valori che verrebbero assegnati alle variabili *n*, *n_ori* e *n_ver* della classe *Cell*, le quali a questo punto della modellazione del cruciverba non hanno ancora alcuna rilevanza, per cui vengono poste tutte a 0. Una volta completato l'inserimento di tutti i dati nel vettore, viene inviata una richiesta a *Manager* per l'esecuzione del vero e proprio salvataggio del file, passando ad esso tutte le informazioni raccolte, il quale mette a disposizione un metodo in grado di scriverle su file seguendo una particolare formattazione che poi lo stesso *Manager* sia in grado di comprendere nel momento in cui l'utente desidera riprendere il lavoro sul cruciverba salvato caricandolo. Per prima cosa viene creato il file con il nome e nel percorso specificati dall'utente, quindi avviene la scrittura di tutti i dati: innanzitutto viene registrato in che fase della progettazione il lavoro è giunto, ovvero se il salvataggio viene eseguito dalla prima o dalla seconda schermata dell'interfaccia grafica; in seguito sono inserite le due dimensioni del cruciverba, seguite dall'elenco di tutte le caselle con le informazioni ad esse relative. È previsto inoltre l'inserimento di ulteriori dati, ovvero l'elenco delle parole che compongono il cruciverba, ognuna con la sua definizione ed altre informazioni che la descrivono, ma tali dati saranno disponibili soltanto se il salvataggio ha luogo mentre l'utente sta lavorando nella seconda schermata dell'interfaccia grafica, quindi essi non appariranno in questo caso. E così si conclude il processo di salvataggio. Un'ultima nota da riportare riguardante questo processo è la capacità di notificare all'utente, tramite un *Dialog* informativo, il fallimento della scrittura del file, qualsiasi sia la causa del problema;

- *Aprire un cruciverba precedentemente salvato*: questa operazione è raggiungibile tramite la voce "Apri" nel menù "File" di cui è dotata questa prima schermata dell'interfaccia grafica e l'interazione dell'utente con tale comando provoca l'esecuzione di un metodo che dà inizio ad un processo inizialmente molto simile a quello visto per il salvataggio del cruciverba. Per prima cosa viene costruito e

visualizzato un *Dialog* uguale a quello mostrato nell'operazione analizzata precedentemente, tramite il quale l'utente ha la possibilità di esplorare cartelle e file presenti sul calcolatore fino a trovare e selezionare quello in cui è stato salvato il cruciverba che si desidera aprire. Come in precedenza, anche in questo caso il *Dialog* propone un percorso predefinito ben preciso nel quale cercare il file desiderato, ovvero la cartella "save" fornita insieme all'applicazione, la stessa che veniva proposta in fase di salvataggio. Una volta selezionato il file desiderato, il processo prosegue inviando una richiesta a *Manager* con allegato anche il riferimento al file stesso, alla quale risponde un metodo con il compito di aprirlo e leggerne e interpretarne il contenuto. Esso si aspetta di rilevare sicuramente dati sulla fase di lavorazione in atto al momento del salvataggio (in quale delle prime due schermate dell'interfaccia grafica ci si trovava), sulle dimensioni del cruciverba e su tutte le caselle che lo compongono, mentre opzionalmente possono essere registrate sul file anche delle informazioni riguardanti le parole che compongono il cruciverba e le loro definizioni (questo solo nel caso il salvataggio sia avvenuto nella seconda schermata dell'interfaccia). Non appena lette la fase e le dimensioni del cruciverba, il metodo si occupa di creare una matrice nella stessa maniera in cui veniva costruita all'avvio della precedente versione dell'applicazione, inserendo quindi un'istanza della classe *Cell* in ogni posizione, e successivamente, durante la lettura dei dati sulle singole caselle, riporta questi ultimi in tale matrice. Similmente viene creato un vettore nel quale verranno inserite le informazioni riguardanti le parole che compongono il cruciverba, sempre tali dati siano contenuti nel file, ma questo aspetto verrà osservato meglio in seguito, poiché è strettamente legato all'analisi della seconda schermata dell'interfaccia grafica. Terminata la lettura del file e la memorizzazione delle informazioni in esso contenute, il compito del metodo di *Manager* non è ancora concluso, poiché è necessario anche mostrare graficamente tutti i dati raccolti dal file inserendoli automaticamente nell'interfaccia grafica, in modo tale da consentire all'utente di riprendere la lavorazione esattamente dove si era conclusa. A tal fine il metodo svolge alcune ultime operazioni prima di concludere il processo di apertura del cruciverba: per prima cosa comunica a quella appropriata delle due schermate dell'interfaccia grafica, scelta a seconda della fase di progettazione durante la quale è stato eseguito il salvataggio, valore della quale è stato appreso dal file appena letto, tutti i dati recuperati e richiede ad essa l'esecuzione di un metodo di cui entrambe sono dotate, il quale ha il compito di mostrarli graficamente proprio come se l'utente

stesso li avesse inseriti. Tale metodo denominato “draw”, pur essendo presente in entrambi gli oggetti che definiscono le due schermate dell’interfaccia, si comporta in modo diverso tra le due vista la differenza tra di esse. Quello relativo alla prima schermata in analisi svolge le seguenti operazioni: inizia con il rendere ogni casella che compone la griglia grafica del cruciverba priva di testo, dallo sfondo bianco e abilitata; quindi imposta i Combo Box che indicano le dimensioni del cruciverba ai valori letti dal file, causando l’attivazione del processo già descritto che avviene interagendo con tali controlli e quindi la disabilitazione delle caselle in eccesso; infine viene sfogliata ogni casella entro le dimensioni appena impostate assegnandole i dati ricavati dal file e memorizzati nella matrice, annerendo quindi le caselle opportune ed inserendo la lettera in esse contenuta laddove specificata, ovvero nel caso in cui ve ne fosse una digitata dall’utente al momento del salvataggio del cruciverba. In questo modo si conclude l’aggiornamento dei contenuti dell’interfaccia grafica ed in seguito a ciò il processo si occupa infine di mostrare la schermata dell’interfaccia appropriata, ovvero quella specificata nel file letto, la stessa che si è appena provveduto a preparare per la visualizzazione tramite l’inserimento di tutti i dati necessari, ciò chiaramente rimuovendo l’altra schermata eventualmente mostrata al momento dell’esecuzione del comando “Apri”. L’ultima nota da citare è, similmente al processo di salvataggio visto in precedenza, la capacità dell’applicazione di rilevare un qualsiasi problema verificatosi nel corso di tutte le operazioni appena descritte (come ad esempio una lettura non riuscita del file da aprire) e, nel qual caso, di interrompere il processo e notificare l’accaduto all’utente tramite un *Dialog* informativo;

- *Chiudere l’applicazione*: per svolgere tale operazione vi è un apposito pulsante tra i controlli sottostanti la griglia del cruciverba, ma anche la voce “Esci” nel menù “File” presentato dalla schermata dell’interfaccia grafica;
- *Procedere alla fase successiva della creazione del cruciverba*: per fare ciò è necessario premere il pulsante apposito tra quelli presenti nell’interfaccia di controllo della schermata; in tal modo si avvia un processo che si occupa di raccogliere tutti i dati inseriti dall’utente, di memorizzarli, e di passare alla seconda schermata dell’interfaccia grafica. In dettaglio, per prima cosa vengono analizzate tutte le caselle del cruciverba per assicurarsi che in ognuna di esse sia contenuta esattamente una lettera (con l’esclusione di quelle annerite). Se questa condizione non è rispettata il processo viene interrotto e all’utente viene notificato il problema tramite un *Dialog*

informativo, mentre nel caso opposto si procede con la raccolta dei dati che vengono immagazzinati in un vettore di stringhe in maniera del tutto simile a quanto già visto nel caso del salvataggio del cruciverba in lavorazione, ma stavolta con l'esclusione dei dati riguardanti le definizioni. In breve ogni posizione del vettore conterrà l'identificativo di una casella e la lettera in essa digitata ('*' nel caso si tratti di una annerita), le due informazioni separate da un carattere apposito. A questo punto tale vettore, insieme ai valori di altezza e larghezza del cruciverba, viene passato a *Manager*, il quale avvia un processo di elaborazione dei dati ottenuti: viene creata una matrice esattamente come accadeva nella precedente versione dell'applicazione e come già visto in caso di caricamento di un cruciverba precedentemente salvato, contenente in ogni suo spazio un'istanza della classe *Cell*, la quale a sua volta è descritta dalle informazioni su una singola casella ricavate dal vettore. È chiaro che le variabili *n*, *n_ori*, *n_ver*, *ori* e *ver* verranno ad avere per ora il valore privo di significato predefinito, poiché ancora non si hanno informazioni su di esse. Dopo aver completato la creazione della matrice, *Manager* si occupa di assegnare la numerazione alle caselle che originano una o due parole tra tutte quelle che compongono il cruciverba. Tutte le caselle vengono analizzate e quelle che devono essere numerate vengono identificate nel seguente modo: la casella immediatamente precedente ad esse deve essere annerita (oppure non deve esservene alcuna poiché esse si trovano sul bordo), inoltre devono avere almeno una casella successiva (non devono quindi essere sul bordo) e quest'ultima deve essere normale, in modo tale da comporre una parola di almeno due lettere. Questo controllo viene svolto due volte per ogni casella, prima seguendo la direzione orizzontale e poi quella verticale. Sfogliando tutte le caselle ordinatamente, non appena se ne rileva una che soddisfi le condizioni appena elencate, le si assegna un numero di definizione (quindi il valore della variabile *n*), scelto in ordine crescente partendo chiaramente da uno. Alle variabili *n_ori* e *n_ver* viene assegnato in questo passaggio lo stesso valore appena impostato per *n* nel caso in cui le condizioni appena descritte si verificano nella relativa direzione. Il processo prosegue impostando in maniera adeguata le variabili *n_ori* e *n_ver* delle caselle in cui esse sono rimaste al valore nullo predefinito (con l'esclusione chiaramente delle caselle annerite): ciò si effettua molto semplicemente sfogliando ancora una volta le caselle nel corretto ordine e, per ognuna di esse che presenta almeno una delle due variabili con valore nullo, viene assegnato a tale o tali variabili lo stesso valore che la stessa variabile presenta nella casella precedente

(chiaramente nella corretta direzione), indicando in tal modo che tutte le caselle che compongono una certa parola orizzontale saranno associate alla stessa definizione orizzontale. Infine *Manager* svolge un'ultima operazione: si occupa di individuare tutte le parole che compongono il cruciverba sfogliando tutte le caselle e, nel momento in cui ne trova una numerata, di ricostruire la o le parole che hanno origine da essa. Per fare ciò vengono lette le lettere inserite nelle caselle adiacenti successive a quella individuata seguendo la o le direzioni per le quali nella casella di origine le variabili *n_ori* o *n_ver* hanno lo stesso valore di *n*, il che sta ad indicare che una parola ha effettivamente origine da lì seguendo la relativa direzione. Per ogni parola trovata in questo modo viene creata da *Manager* un'istanza della classe *Word*, ovvero l'unità base da inserire in un vettore che andrà a costituire il database che sarà molto importante nell'ambito della seconda schermata dell'interfaccia grafica. Ogni istanza di *Word* viene descritta dalla parola stessa per la quale viene creata, la direzione in cui è orientata all'interno del cruciverba ed il numero assegnato alla casella da cui essa ha origine; nessun dato è ancora disponibile per quanto riguarda la definizione ad essa associata. Terminata questa operazione e conclusasi quindi la costruzione della matrice e del vettore delle parole che compongono il cruciverba, il controllo torna al metodo avviato in origine dall'utente alla pressione del pulsante per passare dalla prima alla seconda schermata dell'interfaccia grafica dell'applicazione. Ciò che si occupa di fare prima di effettuare l'effettivo passaggio alla seconda schermata è il visualizzare un *Dialog* che permette all'utente di scegliere se inserire o meno le parole utilizzate nel cruciverba nel file dizionario dei suggerimenti. Nel caso di risposta affermativa viene inviata un'altra richiesta a *Manager*, questa volta riguardante l'aggiornamento del dizionario. Esso risponde raccogliendo tutte le parole utilizzate nel cruciverba dal vettore creato poco prima ed inviando a sua volta delle richieste alla classe *Dictionary*: per prima cosa vengono passate ad essa tutte le parole in modo che possano essere aggiunte a quelle del dizionario se non ancora presenti, operazione svolta nel medesimo modo già visto nella precedente versione dell'applicazione, quindi si richiede di effettuare l'ordinamento alfabetico di tutti i vocaboli, utilizzando anche in questo caso l'algoritmo quicksort nello stesso modo visto in precedenza, ed infine la scrittura del file dizionario in uso con tutti i vocaboli ordinati, inclusi quelli appena aggiunti, operazione anch'essa svolta senza alcuna differenza rispetto alla precedente versione dell'applicazione. Ancora una volta il controllo torna al metodo originario eseguito

alla pressione del pulsante per procedere nella modellazione del cruciverba in costruzione, e di nuovo esso inoltra una richiesta a *Manager*, l'ultima, ovvero quella che prevede l'effettivo passaggio alla seconda schermata dell'interfaccia grafica. In questo caso *Manager* si occupa di nascondere la finestra alla quale si è lavorato finora e di preparare la seconda prima di renderla effettivamente visibile e pronta a gestire l'interazione con l'utente.

– **Funzionalità della seconda schermata:**

- *Inizializzazione*: come detto poco prima, prima di rendere questa seconda schermata visibile e quindi pronta a gestire l'interazione con l'utente, vi è un'operazione che *Manager* richiede ad essa di portare a termine, ovvero visualizzare le parole che compongono il cruciverba nelle due liste che costituiscono il pannello principale della schermata stessa, quella delle parole con definizione e quella delle parole ancora prive di definizione. Ciò è ottenuto creando due nuovi vettori di stringhe, il contenuto dei quali sarà lo stesso che verrà visualizzato nelle due liste. Sfolgiando tutte le parole contenute nel database, a seconda se presentano una definizione o meno, ognuna di esse viene copiata all'interno di uno dei due vettori ed infine viene fatto assumere alle due liste grafiche il contenuto di tali vettori. Da notare che nel caso delle parole con una definizione già associata, quest'ultima viene riportata nella stringa accanto alla parola a cui appartiene, in modo che anch'essa sia visibile nella lista grafica di destra, quella che mostra appunto i vocaboli con una definizione già ad essi associata. Fatto ciò, la schermata viene visualizzata e l'utente può iniziare ad usufruire delle funzionalità da essa offerte.
- *Selezionare una parola da una delle liste*: quando l'utente clicca su una qualsiasi delle parole elencate in una delle due liste mostrate dall'interfaccia grafica per poi poter lavorare sulla definizione ad essa relativa, il gestore degli eventi della lista interessata avvia l'esecuzione di un metodo che deve essere in grado di visualizzare tutti i dati relativi alla parola selezionata. Per prima cosa tale metodo provoca la deselegione della parola eventualmente selezionata nell'altra lista, in modo tale che in ogni caso quello appena scelto dall'utente sia l'unico vocabolo selezionato tra tutti quelli elencati in entrambe le liste, quindi si occupa di eseguire un altro metodo in grado di effettuare tutte le altre operazioni necessarie. Innanzitutto esso registra la parola selezionata in una variabile, in modo tale da renderla immediatamente visibile a tutti i metodi che dovessero avere bisogno di conoscerla; quindi va a cercare la parola stessa all'interno del vettore database con l'elenco di tutte quelle che

compongono il cruciverba, in modo da poter ricavare tutti i dati ad essa relativi e poterli visualizzare negli appositi componenti dell'interfaccia grafica. In particolare la definizione, se presente, viene stampata nell'apposito campo di testo, in modo tale da poter essere letta, modificata o cancellata dall'utente, mentre il numero della casella numerata nella quale la parola ha origine e la direzione in cui è orientata all'interno del cruciverba vengono riportate nell'etichetta che accompagna il campo di testo della definizione. In questo modo viene data all'utente una visione completa dello stato della parola selezionata: le sue coordinate (ad esempio "1 orizzontale") seguite dalla definizione ad essa assegnata (qualora presente);

- *Inserire una definizione:* una volta selezionata una qualsiasi parola da una delle due liste mostrate dall'interfaccia grafica, l'utente ha la possibilità di digitare una definizione per tale parola nel campo di testo apposito, oppure di modificare o cancellare quella già esistente. Qualunque sia il contenuto del campo di testo, esso sarà memorizzato nel database al momento della pressione dell'adiacente pulsante per il salvataggio della definizione da parte dell'utente. Al manifestarsi di tale evento verrà infatti eseguito un metodo con il compito di inserire la stringa presente nel campo di testo come definizione relativa alla parola attualmente selezionata, anche nel caso ciò significasse privare quest'ultima della definizione già presente in precedenza, ovvero nel caso in cui l'utente cancelli il testo presente nel campo e prema a quel punto il pulsante per il salvataggio delle modifiche operate. In seguito alla memorizzazione del testo della definizione, il metodo si occupa di richiamarne un altro con il compito di svolgere un'altra operazione, già vista al momento dell'inizializzazione di questa finestra: si tratta di quello che aggiorna i due vettori contenenti rispettivamente i vocaboli con assegnata una definizione o meno e di seguito anche le due liste grafiche, rispecchiando il contenuto dei vettori appena modificati. Ciò è necessario poiché l'utente potrebbe aver inserito una definizione relativa ad una parola che prima ne era priva o viceversa, in modo tale da rendere necessario uno spostamento da una lista all'altra del vocabolo in questione;
- *Visualizzare le informazioni riguardanti l'applicazione:* effettuare questa operazione nella seconda schermata dell'interfaccia grafica non ha la minima differenza rispetto a quella precedente: viene visualizzata la finestra prodotta dall'istanza della classe *About* all'interazione dell'utente con la voce "About" nel menù "Help" presentato dall'interfaccia;

- *Visualizzare le istruzioni per l'uso delle funzionalità della prima schermata:* anche per questa operazione vale lo stesso detto per la visualizzazione della finestra delle informazioni, ovvero il funzionamento è precisamente lo stesso già descritto per la prima schermata dell'interfaccia grafica. L'unica differenza è che all'interazione dell'utente con la voce "Help" nel menù omonimo, la classe di cui viene costruita un'istanza che andrà a generare la finestra è ora *DefinitionsHelp*, in tutto simile alla *CrosswordHelp* ma con un contenuto testuale differente, ora riferito a questa seconda schermata dell'interfaccia;
- *Impostare le proprie preferenze:* ancora una volta non vi è nessuna differenza per quanto riguarda questa operazione tra la prima e questa seconda schermata dell'interfaccia grafica. All'interazione dell'utente con la voce "Preferenze" nel menù "Opzioni" viene creata e visualizzata la finestra ricavata dalla classe *Options*, la quale consentirà di attuare le stesse scelte viste in precedenza nelle medesime modalità;
- *Salvare in un file il dizionario in lavorazione:* anche in questo caso si rimanda all'analisi della precedente schermata per il funzionamento dettagliato di questa opzione. Vi sono però delle differenze sostanziali che devono essere descritte: esse riguardano chiaramente la raccolta dei dati che verranno poi comunicati a *Manager* per l'effettiva registrazione sul file scelto tramite l'apposito *Dialog*. Ora infatti viene ancora costruito un vettore di stringhe ognuna contenente tutte le informazioni su una casella, ma tali dati non vengono più ricavati direttamente dalla modellazione del cruciverba operata dall'utente tramite l'interfaccia grafica, ma bensì dalla matrice che già li raccoglie tutti in maniera ordinata e chiara, incluse le variabili riguardanti la numerazione e le definizioni relative ad ogni casella. Inoltre ora viene costruito anche un secondo vettore, quello relativo alle parole che compongono il cruciverba che nella prima schermata non aveva ancora ragione di esistere. Anche questa struttura dati contiene delle stringhe, ognuna delle quali è composta dalla parola stessa seguita da un numero che indica il suo orientamento all'interno del cruciverba (0 in caso di senso orizzontale e 1 in caso di senso verticale), quindi un altro che indica il numero della definizione associata alla parola ed infine la definizione stessa; ogni elemento separato ovviamente da un apposito carattere. Per quanto riguarda le parole ancora prive di definizione, quest'ultima sarà sostituita nella stringa da un asterisco '*'. Tutti i dati così raccolti vengono quindi passati a *Manager* che si occuperà della scrittura del file così come è già stato descritto in precedenza,

inserendovi questa volta anche le informazioni presenti nel vettore delle parole assente nel caso di salvataggio effettuato dalla prima schermata dell'interfaccia. Anche questi nuovi dati vengono scritti nel file rispettando una determinata formattazione che poi *Manager* sarà in grado di interpretare nel momento del caricamento del cruciverba che si sta salvando.

- *Aprire un cruciverba precedentemente salvato*: per quanto riguarda questa operazione, non vi è alcuna differenza di implementazione tra essa e quella già vista per la schermata precedente. Viene quindi mostrato il *Dialog* tramite il quale l'utente può scegliere il file da aprire, quindi *Manager*, a seconda dei dati letti ed in particolare da quale delle due schermate è stato effettuato il salvataggio, recupera i dati da esso scritti in precedenza nel file ricostruendo la matrice con le informazioni sulle caselle del cruciverba e, qualora presente, il vettore riguardante le parole che lo compongono. Finito ciò, viene visualizzata quella appropriata tra le due finestre dell'interfaccia grafica una volta aggiornata tramite l'inserimento dei dati appena memorizzati nelle strutture dati, tutto esattamente come già visto in precedenza per la prima schermata;
- *Visualizzare la struttura del cruciverba*: alla pressione da parte dell'utente del pulsante incaricato di effettuare tale operazione, il gestore degli eventi esegue un metodo che si occupa di creare un'istanza di una classe denominata *MiniCrossword*, facente parte del *presentation tier* all'interno dell'architettura *three-tier* dell'applicazione, in quanto essa rappresenterà una finestra che verrà visualizzata e che mostrerà la griglia del cruciverba così come è stata modellata nella prima schermata dell'interfaccia grafica. Ad essa viene comunicata la matrice contenente tutti i dati riguardanti le caselle che compongono il cruciverba e le dimensioni di quest'ultimo, in maniera tale da permetterle di riprodurre graficamente l'intera griglia completa di lettere e caselle annerite. La finestra presenta unicamente un pannello nel quale verrà riprodotto il cruciverba e, sotto ad esso, un pulsante per la chiusura della finestra una volta terminata la sua consultazione. In fase di costruzione vengono inserite nel pannello principale tante caselle grafiche (campi di testo di forma quadrata) quante indicate dalle dimensioni che il cruciverba deve avere, tutte quante impostate come disabilitate, poiché lo scopo di questa finestra è puramente informativo e non prevede alcuna interazione con l'utente (se non la chiusura della finestra stessa alla pressione dell'apposito pulsante); fatto ciò in ognuna di esse viene inserita la lettera ad essa corrispondente, dato ottenuto tramite

l'analisi della matrice con le informazioni di ogni casella, oppure viene colorata di nero nel caso debba essere annerita;

- *Visualizzare i suggerimenti da dizionario online*: alla pressione del pulsante apposito, il gestore degli eventi si occupa di eseguire un metodo che deve essere in grado di aprire la pagina web del dizionario online attivo al momento relativa alla parola selezionata in una delle due liste grafiche, ciò tramite l'utilizzo di un browser installato sul calcolatore in uso. Tale metodo verifica innanzitutto che vi sia effettivamente una parola selezionata tra quelle nelle due liste, altrimenti il processo non potrebbe continuare e l'utente riceverebbe una notifica tramite un *Dialog*, dopodiché lascia che sia *Manager* a svolgere il resto del lavoro, al quale si limita a comunicare la parola selezionata. Per realizzare la funzionalità appena descritta *Manager* si appoggia ad una libreria open source esterna denominata *BrowserLauncher2*²⁰, giunta al momento della stesura di questo testo alla sua versione 1.3, la quale mette a disposizione degli strumenti per interagire con i browser installati sul computer in uso. La principale operazione che è possibile svolgere tramite essa è proprio la costruzione di un oggetto in grado di aprire di una pagina web tramite un browser (che può essere quello predefinito del computer in uso oppure uno a scelta dell'utente) passandogli il link desiderato, impostando facoltativamente ulteriori opzioni, come la visualizzazione della pagina in una nuova istanza del browser (invece che ad esempio in una nuova scheda). La versione di questa libreria utilizzata nell'applicazione fornisce il supporto per tutti i maggiori browser e delle piattaforme Windows, Unix/Linux e Mac, alle quali sa interfacciarsi per rilevare informazioni quali appunto il browser predefinito impostato. È da dire che dalla sua versione 6, Java presenta una nuova API, denominata *Desktop*, che rende non più necessaria la libreria *BrowserLauncher2* poiché è in grado di consentire alle applicazioni Java di aprire URI e file con i programmi registrati per tale scopo sulla piattaforma in uso, tra cui quindi anche gli indirizzi web tramite il browser predefinito²¹. Tuttavia l'applicazione è stata implementata utilizzando una versione precedente di Java, quindi senza il supporto della nuova API, e così è stato mantenuto l'utilizzo di *BrowserLauncher2*, tra l'altro perfettamente funzionale ai fini delle necessità dell'applicazione. Tornando a *Manager*, esso rileva l'indirizzo web del dizionario online attualmente in uso e lo compone in maniera appropriata con la

²⁰ Vedere la pagina relativa a *BrowserLauncher2* ospitata da Sourceforge: <http://browserlaunch2.sourceforge.net/>

²¹ Vedere la documentazione online per l'API *Desktop*: <http://java.sun.com/javase/6/docs/api/java/awt/Desktop.html>

parola ricevuta in ingresso, in modo tale da ottenere il link alla pagina relativa alla parola desiderata da aprire tramite browser. A questo punto viene creata un'istanza di *BrowserLauncher*, classe messa chiaramente a disposizione dalla libreria *BrowserLauncher2*, alla quale viene passato tale link finale, e la quale si occuperà di aprire tale link lanciando il browser predefinito installato sul calcolatore in uso. Nel caso in cui si verificano degli errori di qualsiasi genere durante lo svolgimento di queste ultime operazioni, il processo è impostato in maniera tale da notificare tali eventi all'utente tramite un *Dialog* informativo;

- *Chiudere l'applicazione*: similmente a quanto valeva per la precedente schermata dell'interfaccia grafica, per effettuare tale operazione anche in questa è possibile premere l'apposito pulsante tra i controlli localizzati nella parte inferiore della finestra, ma anche interagire con la voce "Esci" nel menù "File" presentato dalla schermata;
- *Procedere alla fase successiva della creazione del cruciverba*: questa operazione è realizzabile tramite la pressione dell'apposito pulsante nella parte inferiore della schermata e porta alla visualizzazione della terza ed ultima finestra tra quelle che guidano l'utente nella creazione del cruciverba in Flash. Il metodo che viene eseguito al verificarsi dell'interazione dell'utente con il pulsante in questione si occupa per prima cosa di effettuare un necessario controllo: viene verificato che tutte le parole in database abbiano una definizione a loro associata, ovvero che l'utente abbia inserito una definizione per ogni vocabolo che compone il cruciverba e che quindi stia attualmente visualizzando ognuno di essi nella lista di destra, quella relativa alle parole con già una definizione ad esse associata. Nel caso in cui vengano rilevati vocaboli ancora privi di definizione, il processo verrà arrestato e l'utente riceverà una notifica dell'accaduto tramite un *Dialog* informativo; altrimenti le operazioni possono proseguire, ed il metodo mette a disposizione l'effettuazione di un salvataggio del cruciverba completato per prevenire la sua perdita nel caso in cui si riscontrino degli errori nel prosieguo del processo di generazione. Questa opzione viene resa disponibile all'utente tramite un *Dialog* che gli consente di effettuare una scelta affermativa o negativa. Nel primo caso il metodo semplicemente avvia lo stesso processo di salvataggio già visto in dettaglio nel momento in cui si interagisce con la voce "Salva" nel menù "File" dell'interfaccia, quindi dalla visualizzazione del *Dialog* per la scelta del file in cui si vuole memorizzare il cruciverba fino all'effettiva scrittura dei dati in esso; nel secondo caso invece questa opzione viene

ignorata ed il metodo si rivolge a *Manager*, il quale completerà il resto del processo. Esso per prima cosa esegue un'operazione che ha il compito di completare il riempimento della matrice delle caselle che compongono il cruciverba con i dati raccolti dalla seconda schermata dell'interfaccia grafica: tutte le istanze della classe *Cell*, e quindi le caselle, in essa contenute vengono sfogliate e vengono assegnati i valori alle loro variabili *ori* e *ver*, ovvero le due definizioni che esse dovranno visualizzare nel momento in cui vengono selezionate nella futura applicazione Flash. Per fare ciò vengono analizzati i valori di *n_ori* e *n_ver* di ognuna: se una di esse avrà valore 0 allora la definizione relativa a quella direzione per la casella in analisi sarà vuota (e nella stringa *ori* o *ver* che la deve contenere viene inserito un asterisco '*'); se invece possiede un valore numerico maggiore di 0, viene avviata una ricerca all'interno del vettore contenente tutte le parole che compongono il cruciverba e quando viene trovata una corrispondenza per quanto riguarda sia l'orientamento di scrittura che il numero associato alla definizione, la stringa della definizione trovata nel vettore viene assegnata alla casella in esame nell'opportuno orientamento. Per fare un esempio prendiamo in esame una casella del cruciverba che si trova nella prima casella della seconda riga. Immaginiamo che essa sia numerata e presenti ad esempio il numero 5, inoltre che almeno la casella alla sua destra e quella sopra non siano annerite. Se analizzassimo le sue variabili troveremmo i seguenti valori: $n = 5$; $n_ori = 5$; $n_ver = 1$; il che sta ad indicare che la casella è tra quelle numerate (in particolare detiene il numero 5) e che quindi almeno una parola ha origine in essa. Si nota poi che la lettera che essa contiene è la prima della parola orizzontale di cui fa parte, e che quindi origina, ma non di quella verticale, la quale ha invece origine nella casella identificata dal numero 1 (che in questo caso specifico sarà chiaramente quella soprastante, ovvero la prima casella del cruciverba). Nel passaggio che si sta analizzando quindi le variabili *ori* e *ver* della casella in analisi verranno a contenere rispettivamente le stringhe delle definizioni associate alle parole con coordinate "5 orizzontale" e "1 verticale" trovate nel vettore dei vocaboli che compongono il cruciverba. Terminato il completamento della matrice tramite il processo appena descritto, *Manager* esegue un'ultima operazione: rende invisibile l'attuale seconda schermata dell'interfaccia grafica ed al suo posto mostra la terza ed ultima, dalla quale sarà possibile concludere la generazione del cruciverba in Flash.

– **Funzionalità della terza schermata:**

- *Digitare un nome per il cruciverba*: come già detto, l'unica operazione da eseguire in questa finestra finale prima dell'effettiva generazione del cruciverba è l'inserimento di un nome da assegnare ad esso, il quale poi andrà ad identificare la directory che verrà creata e nella quale avverrà la costruzione dell'applicazione Flash. Per l'utente è sufficiente digitare tale nome nell'apposito campo di testo, dopodiché è possibile procedere con la generazione del cruciverba senza alcun altro accorgimento, se non quello di leggere le note sulla costruzione dell'applicazione Flash presentate dalla schermata;
- *Chiudere l'applicazione*: a differenza delle due precedenti schermate dell'interfaccia grafica, quest'ultima non è dotata di menù, poiché giunti a questo punto della costruzione del cruciverba non è più richiesta alcuna funzionalità particolare se non la sua sola effettiva generazione. Rimane tuttavia chiaramente la possibilità di chiudere l'applicazione tramite l'apposito pulsante tra i due presentati da questa schermata;
- *Generare il cruciverba in Flash*: questa operazione è sostanzialmente invariata rispetto a quella vista nella precedente versione dell'applicazione. Alla pressione del pulsante per avviare la generazione viene eseguito un metodo che innanzitutto verifica che sia stato digitato un nome per il cruciverba nell'apposito campo di testo (in caso contrario l'utente viene avvisato tramite un *Dialog*), quindi si assicura che non vi sia già una directory con lo stesso nome appena scelto dall'utente (e di nuovo in caso contrario un *Dialog* metterà l'utente al corrente del problema), dopodiché invia a *Manager* la richiesta di creare un'istanza di *Generator*, ovvero la classe che include tutti gli strumenti necessari per la realizzazione della generazione dell'applicazione Flash. *Manager* esegue, comunicando a tale nuovo oggetto tutte le informazioni necessarie, ovvero la matrice con le informazioni per ogni casella del cruciverba, le dimensioni di quest'ultimo ed il nome scelto dall'utente per esso. A questo punto vengono richiamate le funzionalità di *Generator* già descritte per la precedente versione dell'applicazione, le quali si occupano di creare la cartella con il nome scelto dall'utente, di copiarvi tutti i file necessari per il funzionamento dell'applicazione Flash, ed infine di crearvi i due file principali (il documento XML e la pagina HTML) che vengono generati. Come accadeva anche in precedenza, se in qualsiasi fase del processo viene riscontrato un errore, questo sarà notificato all'utente tramite un *Dialog* informativo. Viene mostrato all'utente un *Dialog* anche

nel caso in cui l'operazione viene conclusa senza alcun problema, in questo caso chiaramente notificando l'esito positivo della costruzione dell'applicazione Flash.

Queste appena elencate sono le funzionalità presentate dall'applicazione in un primo momento successivo alla sua riprogettazione. Ben presto, dopo i primi test, sono state ideate però alcune ulteriori modifiche da applicarvi ed opzioni da aggiungervi per renderla più corretta, completa ed interessante.

5.8. Le ultime ulteriori modifiche

Qui vengono elencati gli ultimi aggiustamenti e funzionalità implementati ed applicati all'applicazione Java, completati i quali essa è stata ritenuta giunta alla sua revisione finale. Ecco dunque le modifiche ideate:

- **Modificare la funzionalità riguardante l'annerimento delle caselle:** in effetti la gestione dell'annerimento delle caselle finora visto nell'ambito della prima schermata dell'interfaccia grafica presenta un problema che rende la scorciatoia tramite click con il tasto destro del mouse sulla casella interessata non solo preferibile rispetto alla pressione del pulsante apposito tra quelli nell'interfaccia di controllo, ma anche l'unica effettivamente funzionante in ogni situazione. Il problema sorge nel momento in cui si desidera ripristinare una casella precedentemente annerita tramite l'utilizzo del pulsante dell'interfaccia piuttosto che il click con il tasto destro del mouse: per fare ciò è necessario che la casella da ripristinare sia selezionata, ma ciò è impossibile, in quanto se annerita una casella è anche disabilitata e quindi non selezionabile. In altre parole tramite il pulsante grafico è possibile annerire una casella normale, ma non è possibile effettuare l'operazione inversa, a meno che non la si effettui prima di selezionare una qualsiasi altra casella. Si è deciso quindi di modificare completamente il funzionamento del pulsante grafico: ora premendolo si congela lo stato di ogni casella e l'unica operazione possibile per l'utente è l'annerimento delle caselle effettuabile con un semplice click del tasto sinistro del mouse su quelle interessate. Per terminare questa "modalità di annerimento" e tornare alla normale lavorazione del cruciverba è sufficiente premere nuovamente il pulsante. Rimane invece invariato il perfettamente funzionale sistema di annerimento attivabile tramite click del tasto destro del mouse;
- **Aggiungere una funzionalità che ricavi automaticamente delle definizioni per la parola selezionata dal dizionario online attivo:** questa funzionalità, relativa chiaramente alla

seconda schermata dell'interfaccia grafica, consiste nell'aggiunta di un pulsante tra i controlli già presenti, alla pressione del quale l'applicazione non si limita ad aprire la pagina web del dizionario online in uso relativa alla parola selezionata, ma bensì si occupa di analizzarla interamente alla ricerca automatica di una o più definizioni presentate. Qualora ve ne siano, esse dovranno essere mostrate all'utente, il quale potrà sceglierne una ed associarla così al vocabolo selezionato tramite un semplice click;

- **Aggiungere una funzionalità che svolga lo stesso compito della precedente ma per tutte le parole ancora prive di definizione che compongono il cruciverba:** questa funzionalità è in tutto simile alla precedente, ma effettua tale procedimento per ogni parola tra quelle ancora prive di definizione, ovvero quelle elencate nella lista di sinistra della seconda schermata dell'interfaccia grafica.
- **Aggiungere una funzionalità che ricavi la traduzione della parola selezionata dal dizionario online attivo in una lingua specificata:** questa funzionalità è stata ideata per potenziare il supporto dell'utilizzo dell'applicazione in ambiti didattici di tipo linguistico. Svolge le stesse operazioni di quella sopra descritta, ma invece delle definizioni essa ricerca eventuali traduzioni all'interno della pagina del dizionario online, ed in particolare quelle relative ad uno specifico linguaggio che viene fatto scegliere all'utente. Il supporto per i dizionari online dell'applicazione si limita ora, come già detto, a Wikizionario, il quale prevede appunto informazioni circa le traduzioni di numerosi vocaboli;
- **Aggiungere una funzionalità che svolga lo stesso compito della precedente ma per tutte le parole ancora prive di definizione (o traduzione) che compongono il cruciverba:** come avveniva per la funzione della ricerca delle definizioni, anche questa per le traduzioni ha la sua variante in grado di ricercarne per tutti i vocaboli che risultano ancora privi di una definizione;
- **Scegliere una delle due nuove modalità di ricerca online:** sono state appena descritte due modalità di ricerca nel dizionario online, ovvero quella per le definizioni e quella per le traduzioni in una lingua specifica, entrambe con la variante per una singola parola e per tutte quelle ancora prive di definizione o traduzione associata. Si è dunque pensato di aggiungere due pulsanti alla seconda schermata dell'interfaccia grafica, ognuno dei quali avrebbe avviato una delle due varianti della ricerca (per una singola parola o per tutte), ma è inoltre necessario un meccanismo per la scelta di quale deve essere l'oggetto della ricerca (definizioni o traduzioni). Ecco che quindi è stata pianificata anche l'aggiunta di un Combo Box che proponga le due tipologie di ricerca tra le quali l'utente può scegliere, ed a seconda dell'opzione selezionata i due pulsanti citati in precedenza agiranno di conseguenza. Un

secondo Combo Box inoltre avrebbe la funzione di permettere all'utente la scelta della lingua alla quale fare riferimento durante la ricerca delle traduzioni.

Quelle appena elencate sono le tre grandi modifiche operate sull'applicazione in quest'ultima fase di elaborazione. Ora verranno analizzate in dettaglio da un punto di vista maggiormente tecnico, come fatto in precedenza per ogni altra funzionalità che caratterizza l'applicazione:

- **Annerire le caselle del cruciverba:** il pulsante con il compito di attivare questa funzionalità ed il metodo che viene eseguito alla sua pressione vengono interamente modificati. Innanzitutto il pulsante stesso diventa ora un interruttore che presenta quindi due possibili stati (attivato o meno), ognuno dei quali rappresenta una diversa modalità di lavorazione al cruciverba; è possibile spostarsi da uno stato all'altro premendo sul pulsante. Nel momento in cui l'interruttore viene attivato queste sono le operazioni svolte: innanzitutto viene posto a "true" il valore di una variabile booleana appositamente creata, la quale sta ad indicare se la modalità di annerimento delle caselle è attiva o meno; poi vengono disabilitate tutte le caselle non annerite e gli altri controlli dell'interfaccia che consentono l'interazione con il cruciverba, ovvero i pulsanti per il cambio di direzione di scrittura, per ottenere suggerimenti dal dizionario e per stampare la griglia del cruciverba, ed i Combo Box per l'impostazione delle dimensioni del cruciverba, oltre al pulsante per proseguire verso la seconda schermata dell'interfaccia grafica. Mentre l'interruttore è attivato in sostanza l'utente può esclusivamente interagire con il cruciverba esclusivamente cliccando sulle caselle con il mouse. A questo punto va fatto notare che anche il metodo che gestisce gli eventi di click con il mouse sulle caselle è stato modificato, ed in particolare esso ora prevede un'opzione aggiuntiva rispetto a prima: al momento del click dell'utente esso osserva il valore della variabile booleana che indica se la modalità di annerimento è attiva o meno e, nel primo caso, si occupa di colorare in nero la casella interessata qualora fosse bianca oppure il contrario qualora essa fosse già di colore nero. Nel caso in cui invece ci si trovi nella modalità normale, il comportamento di tale metodo è identico a prima. Quando l'utente interagisce con l'interruttore disattivandolo, ciò di cui si occupa il metodo eseguito al verificarsi di tale evento è l'impostazione a "false" del valore della variabile booleana indicante la modalità attiva, la riattivazione di tutte le funzionalità dell'interfaccia grafica e delle caselle che in quell'istante presentano una colorazione bianca;
- **Scegliere una delle due nuove modalità di ricerca online:** innanzitutto questa funzionalità prevede l'inserimento di un nuovo piccolo pannello all'interno della seconda schermata

dell'interfaccia grafica, posto nella parte alta della finestra accanto all'etichetta introduttiva che ricopre il margine nord dell'intera struttura. In tale pannello vengono inseriti quindi due Combo Box, uno per la scelta della modalità di ricerca online da effettuare e l'altro per il linguaggio interessato nel caso l'utente scelga di ricercare traduzioni, accompagnati da opportune etichette che li introducano. All'interazione dell'utente con il primo Combo Box, ovvero quello che permette la scelta tra la modalità di ricerca di definizioni oppure traduzioni, il metodo ad esso associato si occupa di impostare il valore di una variabile booleana appositamente creata con il compito di indicare tramite il valore "false" la prima delle due modalità e tramite "true" la seconda; quindi viene disabilitato il secondo Combo Box riguardante la scelta del linguaggio per le traduzioni nel caso l'utente abbia selezionato la ricerca di definizioni, oppure al contrario esso viene abilitato nel caso l'utente preferisca le traduzioni. Infine il metodo si preoccupa di ristampare tutte le stringhe che appaiono nell'interfaccia grafica. Il motivo di quest'ultima operazione è che al passaggio da una delle due modalità di ricerca all'altra è previsto di modificare il testo contenuto nelle etichette dei due pulsanti che attivano tali ricerche specificando in tal modo ciò che l'utente otterrà, ovvero se si tratterà di definizioni oppure traduzioni. È quindi chiaro che nel metodo per la stampa delle stringhe testuali viene aggiunto un controllo che porti alla visualizzazione di una stringa diversa in tali etichette a seconda della modalità di ricerca scelta dall'utente, ciò osservando la variabile booleana che la identifica. Per quanto riguarda il Combo Box riguardante il linguaggio in cui ricercare le traduzioni, il metodo che viene eseguito all'interazione con esso da parte dell'utente, accessibile ovviamente solo nel caso esso abbia in precedenza optato per la ricerca di traduzioni, ovvero se il Combo Box stesso è abilitato, si occupa semplicemente di riportare in una variabile appositamente creata il linguaggio scelto tra quelli proposti. Da notare che le lingue supportate dall'applicazione (sono state scelte italiano, inglese e tedesco) vengono rese disponibili alla selezione tramite il Combo Box da del codice inserito nel metodo per la stampa delle stringhe dell'interfaccia grafica. In particolare vengono inserite le lingue diverse da quella scelta per il dizionario online selezionato (che rappresenterà quella di origine). Se ad esempio il dizionario online attivo al momento è il Wikizionario in lingua italiana, allora nel Combo Box verranno inserite le opzioni per la scelta dei soli linguaggi inglese e tedesco e così via;

- **Ottenere delle definizioni o delle traduzioni per la parola selezionata dal dizionario online attivo:** il metodo che viene eseguito alla pressione del pulsante associato a questa funzionalità si occupa innanzitutto di rilevare la parola selezionata dall'utente tra quelle nelle due liste della schermata (nel caso non ve ne sia alcuna selezionata il processo si

interrompe e viene mostrato all'utente un *Dialog* di notifica). Successivamente viene osservato il valore della variabile booleana che sta ad indicare la modalità di ricerca scelta dall'utente (definizioni o traduzioni). A seconda di tale scelta viene inviata una richiesta a *Manager* comunicando ad esso la parola interessata e, nel caso si desiderino le traduzioni, anche la lingua selezionata dall'utente memorizzata in un'apposita variabile come illustrato in precedenza. Qualsiasi sia la richiesta che giunge a *Manager*, inizialmente esso svolge alcune operazioni comuni: innanzitutto vengono inizializzati due vettori, uno dei quali ospiterà le righe di codice HTML della pagina web del dizionario online attivo relativa alla parola selezionata, mentre l'altro conterrà tutte le definizioni o le traduzioni trovate. Successivamente viene ricostruito l'URL della pagina web che dovrà essere letta basandosi sulle informazioni riguardanti il dizionario online attivo e sulla parola che si sta ricercando. Ottenuto l'URL la pagina web viene sfogliata riga per riga, memorizzando ognuna di queste ultime in uno dei due vettori creati precedentemente per poterle successivamente analizzare in dettaglio. Tuttavia se durante questo processo viene rilevata in una qualsiasi riga tra quelle che compongono la pagina web una particolare stringa che sta ad identificare l'assenza di tale pagina all'interno del dizionario online (se ad esempio la parola ricercata non è ancora stata inclusa tra tutte quelle che compongono tale dizionario), allora il processo viene immediatamente interrotto e come al solito l'utente verrà a conoscenza del problema tramite un *Dialog* che lo notifica. Terminata la memorizzazione delle righe che compongono la pagina web nel vettore apposito, *Manager* inizia ad analizzarle una per una alla ricerca di definizioni o traduzioni, ed a questo punto chiaramente il processo si diversifica a seconda di quale delle due cose si stia ricercando. A seconda di ciò e di quale sia il dizionario online attivo, tale operazione richiede diversi tipi di analisi basati su particolari tag HTML o stringhe che identifichino gli elementi interessati tra tutte le informazioni contenute nella pagina. Per quanto riguarda la ricerca di definizioni, *Manager* è in grado di rilevare la riga o le righe che ne contengono; in un secondo momento tali righe vengono ripulite da tutti i tag HTML ed altri codici che contengono, isolando in tal modo le pure definizioni, ognuna delle quali viene inserita all'interno dell'apposito vettore creato in precedenza. Similmente accade per le traduzioni, con la principale differenza che in questo caso si prevede di rilevare più di una definizione per ogni riga e quindi si opera isolando le diverse stringhe intervallate da tag HTML presenti in ogni riga, ripulendole poi anche in questo caso da eventuali codici utilizzati dal dizionario online ma privi di interesse ai fini dell'applicazione generatrice di cruciverba. *Manager* conclude quindi il suo compito riempiendo un vettore con tutte le definizioni o le traduzioni trovate, ma il processo non ha ancora termine, infatti il metodo

originario, ottenuto tale vettore, si occupa di mostrare all'utente i risultati ottenuti dalla ricerca nel seguente modo: per prima cosa verifica che il vettore non sia vuoto ed in caso contrario verrà mostrato all'utente un *Dialog* che informa dell'assenza di risultati utili; quindi costruisce un'istanza della nuova classe *OnlineDefinitions* passandogli lo stesso vettore dei risultati ed la parola del cruciverba per la quale sono state ricercate le definizioni o le traduzioni. Tale classe, che rientra nel *presentation tier* dell'architettura *three-tier*, rappresenta una finestra composta principalmente da un'etichetta introduttiva, una lista nella quale verranno visualizzati tutti gli elementi contenuti nel vettore e tramite la quale sarà possibile all'utente selezionare la definizione o traduzione preferita, ed infine due pulsanti, uno per confermare l'inserimento nel database della definizione o traduzione selezionata dall'utente ed uno per annullare l'operazione in corso. Nel momento in cui viene premuto il pulsante "Ok" di conferma, il metodo ad esso associato chiude la finestra, ma non prima di effettuare le seguenti azioni: esso va a rilevare la stringa selezionata tra quelle nella lista (nel caso nessuna lo sia l'utente riceverà una notifica tramite un *Dialog*) e richiede l'esecuzione di un altro metodo tra quelli della classe *Definitions*, ovvero la seconda schermata dell'interfaccia grafica, passandogli tale stringa scelta dall'utente e la parola alla quale deve essere associata. Questo metodo si occupa di memorizzare effettivamente la definizione scelta in tal modo dall'utente nel database, associata alla parola per la quale la ricerca è stata effettuata; dopodiché, per concludere l'intero processo, richiama il metodo già visto in precedenza che si occupa di aggiornare le due liste presenti nella seconda schermata dell'interfaccia grafica, in modo tale che la parola alla quale è stata appena assegnata la definizione venga trasferita nella lista di destra, quella dei vocaboli con una definizione assegnata;

- **Ottenere delle definizioni o delle traduzioni per tutte le parole che ne sono prive dal dizionario online attivo:** chiaramente il metodo che viene eseguito alla pressione del pulsante che attiva questa funzionalità è estremamente simile a quello appena visto che svolge lo stesso processo ma per un solo vocabolo invece che per tutti quelli privi di definizione. L'unica differenza risiede nelle righe di codice iniziali, in cui invece che rilevare la parola selezionata tra quelle nelle due liste mostrate dalla finestra dell'interfaccia grafica, questo metodo raccoglie appunto tutte quelle prive di definizione, ovvero quelle che risiedono nella lista di sinistra. Fatto ciò, esso avvia il processo già visto in precedenza per la prima delle parole raccolte, ma, una volta terminato, lo riavvia per il secondo di tali vocaboli e così via fino ad averli esauriti tutti.

E con queste ultime modifiche ed estensioni l'applicazione ha raggiunto la sua versione definitiva così come appare oggi.

CONCLUSIONE

Concludendo si può dire che l'*Iniziativa Lavagne Interattive* proposta dall'Università degli Studi di Trento su iniziativa della Provincia Autonoma di Trento è un progetto dalle indubbe potenzialità che promettono una notevole evoluzione del sistema scolastico e dei metodi didattici adoperati.

Le difficoltà che sempre si presentano in occasione di cambiamenti così sostanziali non sono di immediato superamento e ciò si è visto anche nelle esperienze europee in quest'ambito già realizzate e studiate, ma i progetti pianificati dagli attori impegnati nell'iniziativa voluta dalla Provincia possono portare con il tempo ad un'ottima integrazione delle lavagne interattive nelle scuole e di poterle sfruttare al pieno delle loro potenzialità. Gli studi ed i lavori in atto a tale scopo, come quelli intrapresi dall'Università degli Studi di Trento, ne sono testimoni.

L'applicazione presentata in questa tesi è stata implementata sulla base dell'utilizzo futuro alla quale è destinata, ovvero uno strumento di facile ed efficiente utilizzo per insegnanti ed eventualmente studenti in grado di produrre dei cruciverba di utilità didattica, i quali devono prestarsi ad una soddisfacente risoluzione interattiva.

Un qualsiasi utente privo di specifiche conoscenze informatiche deve essere in grado di utilizzare il generatore ed i cruciverba da esso prodotti; per questo è stata prestata grande attenzione alla portabilità dell'applicazione, in modo che essa possa funzionare senza problemi sulle principali piattaforme e configurazioni dei calcolatori, ed alla realizzazione dell'interfaccia grafica che ne renda l'utilizzo il più immediato e semplice possibile, rendendo allo stesso tempo disponibili utili ed interessanti funzionalità che permettano di facilitare al massimo la modellazione di un cruciverba.

Le funzionalità basilari che ci si aspetta da un programma per la generazione di cruciverba, come ad esempio la modellazione della griglia stessa tra caselle annerite e dimensioni in larghezza ed altezza e l'inserimento delle parole e delle definizioni ad esse associate, sono state integrate nell'interfaccia grafica nella maniera più chiara e di immediato utilizzo possibile. Oltre ad esse sono state implementate diverse altre funzionalità in modo da rendere più interessante l'applicazione e da fornire validi aiuti e suggerimenti all'utente durante il processo di costruzione del cruciverba. Alcuni esempi sono la ricerca di parole da inserire nella griglia del cruciverba prese da un file dizionario, il supporto del dizionario online Wikizionario per ottenere da esso spunti e suggerimenti riguardanti le definizioni da inserire nel cruciverba e le traduzioni in lingue differenti delle parole in esso contenute, oppure addirittura per ricercarle ed inserirle del tutto automaticamente, e la stampa della griglia del cruciverba.

Per favorire l'utilizzo dei cruciverba da un punto di vista didattico linguistico, non è stata implementata solo la ricerca di definizioni e traduzioni sul dizionario online Wikizionario in italiano, ma bensì anche la possibilità di usufruire delle versioni in lingue differenti dello stesso (di predefinito inglese e tedesco); inoltre l'applicazione consente l'utilizzo di diversi file dizionario, dai quali ottenere suggerimenti circa i vocaboli da inserire nella griglia del cruciverba, scegliendone uno diverso tra quelli disponibili in qualsiasi momento. L'utente può aggiungerne ed utilizzarne quindi di ulteriori in qualsiasi lingua desiderata. Infine l'applicazione supporta l'internazionalizzazione, prevedendo la traduzione in più lingue della sua stessa interfaccia grafica (di predefinito sono fornite le traduzioni italiana ed inglese).

Anche i cruciverba prodotti dal generatore sono progettati in modo tale da rendere la loro risoluzione il più naturale possibile, in modo tale che l'utente debba concentrarsi solo su definizioni e vocaboli e non su dettagli tecnici dovuti ad imperfezioni nell'implementazione che possono rendere scomoda e frustrante la risoluzione del gioco.

I cruciverba consentono la digitazione delle lettere al loro interno e la consultazione delle definizioni nel modo più semplice ed intuitivo possibile, mettono a disposizione inoltre funzionalità utili quali la segnalazione degli errori, la visualizzazione della soluzione del gioco e la cancellazione di tutte le lettere già digitate quando richiesto dall'utente, mantenendo allo stesso tempo un'interfaccia basilare e leggera.

L'augurio è quello di un utile e soddisfacente utilizzo dell'applicazione presentata in questa tesi nell'ambito didattico e scolastico della Provincia Autonoma di Trento e di un positivo e fruttuoso proseguo dell'*Iniziativa Lavagne Interattive*, superando le problematiche legate all'introduzione delle lavagne interattive multimediali ed al conseguente cambiamento dei piani didattici da seguire, fino a portare la qualità dell'istruzione nelle scuole della Provincia ad un livello superiore.

BIBLIOGRAFIA

- *Language Tests at School* (1979), di John W. Oller
- *Attività lessicali per classi ad abilità miste. Due strumenti di facile utilizzo: Eclipse Crossword & Puzzlemaker* (2004), di Paolo Torresan
- Sito web di Didacta, portale sulle idee per l'innovazione didattica <http://www.retecivica.mi.it/didacta>
- *Il gioco come risorsa didattica*, di Giacomo Rota
- *A History of the GUI* (1995), di Jeremy Reimer, Ars Technica
- The Interaction-Design.com Encyclopedia, <http://www.interaction-design.org/encyclopedia/>
- *WIMP Interfaces* (1997), di Ashley George Taylor
- *Interazione uomo-macchina: interfacce creative* (2006), di Marco Infussi
- The Java Tutorials: Graphical User Interfaces, <http://java.sun.com/docs/books/tutorial/ui/>
- A Swing Architecture Overview, <http://java.sun.com/products/jfc/tsc/articles/architecture/>
- The Java Language Environment, <http://java.sun.com/docs/white/langenv/>
- The Java Language Specification, Third Edition, <http://java.sun.com/docs/books/jls/>
- *Object Oriented && Java 5 - II Edizione*, di Claudio De Sio Cesari
- *The Introduction of Interactive Whiteboards into Schools in the United Kingdom: Leaders, Led, and the Management of Pedagogic and Technological Change* (2002), di Derek Glover e David Miller
- *What research says about interactive whiteboards* (2003), pubblicato da BECTA
- *Beyond the wow factor: developing interactivity with the interactive whiteboard* (2005), di Gary Beauchamp e John Parkinson
- *The Interactive Whiteboards, Pedagogy and Pupil Performance Evaluation: An Evaluation of the Schools Whiteboard Expansion (SWE) Project: London Challenge* (2007), di Gemma Moss, Carey Jewitt, Ros Levaèia, Vicky Armstrong, Alejandra Cardini e Frances Castle, con analisi statistiche di Becky Allen, Andrew Jenkins e Maggie Hancock con Sue High, School of Educational Foundations and Policy Studies, Institute of Education, University of London
- Sito web di SMART Technologies, <http://smarttech.com/>

- *Reusable Learning Object Strategy: Designing and Developing Learning Objects for Multiple Learning Approaches* (2003), pubblicato da CISCO Systems Inc.
- *Draft Standard for Learning Object Metadata* (2002), pubblicato da IEEE (Institute of Electrical and Electronics Engineers)
- Sito web di ADL (Advanced Distributed Learning), <http://www.adlnet.gov/>
- *Sviluppo ed implementazione di un sistema integrato CMS-LMS* (2006), relazione di tirocinio di Marco Diener
- Sito web di Adobe Systems Inc., <http://www.adobe.com/it/>
- *The Flash History* (2006), di Rick Waldron
- Sito web dell'applicazione Gaupe, <http://www.gaupe.net/>
- Sito web della libreria Java BrowserLauncher2, <http://browserlaunch2.sourceforge.net/>
- *Using the Desktop API in Java SE 6* (2006), di John O'Conner