Alma Mater Studiorum · Università di Bologna

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea in Scienze di Internet

SISTEMA MULTIMEDIALE PER APPRENDERE IL GIOCO DEGLI SCACCHI

Relatore:
Chiar.mo Prof.
DAVIDE SANGIORGI

Presentata da:
ENNIO MISURACA

Sessione III
Anno Accademico 2010/2011

INDICE

Indice delle Immagini

INTRODUZIONE

Chess Trainer nasce dall'intento di creare una piattaforma WEB di *e-learning* per insegnare il gioco degli scacchi, e fornire così un potente strumento di insegnamento a distanza, dove *"l'apprendista"* sia seguito da un *tutor* che, passo dopo passo, segua il suo processo di apprendimento.

L'ambiente di sviluppo pensato per la progettazione dell'interfaccia grafica e dell'applicazione stessa, si basa su un paradigma di programmazione orientato agli oggetti (*OO*, *Object Oriented*).

Il primo passo effettuato, nella fase di progettazione *OO* di un applicazione , è quello di redarre un documento che esplichi i principali requisiti progettuali da raggiungere e soddisfare.

Dopo un sua attenta lettura ed un'analisi accurata, si è voluto realizzare dei diagrammi, seguendo le direttive di *UML*, che aiutino a seguire meglio le interazioni che avvengono tra i vari *attori* del sistema. Con l'aiuto di *UML*, dunque, ho curato in particolare i seguenti aspetti:

- *Casi d'uso*: indicano la descrizione dei servizi offerti da un sistema e degli attori che interagiscono con esso. *UML* descrive i casi d'uso attraverso i *Case Use Diagram*.
- *Modelli di Dominio*: propriamente un *"domain model"*, in informatica, indica un modello che descrive le entità e le loro relazioni all'intero di un sistema progettato secondo il paradigma *OO*. Per la stesura di questo diagramma (in *UML Class Diagram*) ho utilizzato la tecnica denominata *analisi nome-verbo,* dove i nomi sono potenziali candidati a classi e attributi mentre i verbi sono potenziali candidati a responsabilità di classe.

Si faccia poi attenzione all'architettura interna di Chess Trainer, che è basata su invio di messaggi asincroni fra il *client* (studente o docente) e il *server* (l'applicazione), gestisti da *AJAX*, una tecnologia innovativa che permette di sviluppare applicazione WEB interattive.
AJAX permette lo scambio di dati tra *web browser* e *server* in *background*, consentendo così l'aggiornamento dinamico della pagina senza che quest'ultima sia caricata manualmente dall'utente.
La piattaforma di sviluppo adottata è conosciuta con il nome di *LAMP*, ed è una delle più utilizzate a livello mondiale. Questo acronimo prende il nome dalle iniziali delle componenti che lo descrivono:

- *Linux*: sistema opearivo *unix-like* che ospita il *server*
- *Apache*: *Web Server* di comune utilizzo
- *MySql*: è un sistema software che consente la creazione e la manipolazione efficiente di un *database* di tipo relazionale
- *Php*: linguaggio di *scripting* orientato agli oggetti.

LAMP, come anche *AJAX*, non è una tecnologia individuale, ma indica un gruppo di tecnologie utilizzate insieme per lo sviluppo ottimale di un' applicazione WEB.[1]

Un ulteriore aiuto allo sviluppo di Chess Trainer è stato apportato dall'utilizzo di *framework* come *JQuery* e *JQuery UI*, librerie di funzioni *JavaScript* che, attraverso l'utilizzo di metodi e funzioni *built-in*, semplificano enormemente la programmazione lato *client* di una pagina *HTML*. Infatti questi *framework* forniscono metodi per creare, manipolare o eliminare qualsiasi elemento in una pagina web in modo dinamico; gestiscono inoltre ogni evento scatenato dal

1 http://it.wikipedia.org/wiki/AJAX

mouse o dalla tastiera, e controllano le azioni scaturite dall'invio di dati da parte di un *form*.

E' necessario indagare anche i metodi della didattica che diano una linea guida ed una giusta impostazione al nostro percorso; tutti i corsi online sono infatti caratterizzati da forme e metodologie di apprendimento ad alto coinvolgimento e altamente interattive.

Al fine poi di soddisfare i requisiti funzionali di *interattività* e *multimedialità* - che risultano fondamentali in una piattaforma di e-learning -, oltre a fornire una tavola da gioco estremamente dinamica dove ogni pezzo può essere mosso liberamente al suo interno con la tecnica del *drag and drop*, la nostra applicazione, è stata arricchita da tre *tools* di comunicazione telematica, che rendono ogni lezione maggiormente interessante e piacevole:

- la *chat*: strumento fondamentale per un corretto svolgimento della lezione, che permette a studenti e docente di comunicare liberamente durante le fasi di gioco. Grazie ad esso l'insegnante può fornire facilmente delucidazioni o spiegazioni allo studente.
- il *pennarello*: in questa modalità la scacchiera si trasforma in una vera e propria lavagna virtuale grazie all'elemento *canvas*, che è una estensione dell'*HTML* standard che permette il *rendering* dinamico di immagini *bitmap* gestibili attraverso un linguaggio di *scripting*.
- lo *strumento evidenzia casella*: è un tool che aumenta l'interattività del nostro sistema; consiste in una modalità che permette al docente, con un semplice click, di opacizzare il colore in background di ogni casella, rendendola più visibile al discente.

Ultimo strumento di supporto alla lezione, ma non per questo meno importante, è *"Edit Board"* che permette di personalizzare la scacchiera; con questo *tool* è possibile muovere qualsiasi pezzo in qualunque casella libera, cancellare o aggiungere un nuovo elemento o promuovere una pedina. La sua funzionalità è duplice, in quanto oltre a permettere la gestione di mosse speciali (*arrocco*, *en passant* e *promozione*), fornisce anche una potente strumento che dà l'opportunità al docente di creare particolari strategie di attacco o di difesa da mostrare all'alunno in qualsiasi momento della lezione.

Obiettivo fondamentale della teledidattica, oltre a quello di fornire un sistema interattivo e dinamico nel quale comunicare, è consentire la possibilità di creare contenuti utili e di organizzarli secondo gli obbiettivi formativi e le necessità dell'utenza.[2] Secondo tale principio, i docenti/tutor possono salvare intere lezioni o momenti peculiari di gioco arricchiti da commenti e note per ogni mossa o spostamento.

Così facendo vengono creati dei materiali didattici e fruibili per ogni studente in qualsiasi momento e da qualunque supporto collegato alla rete, sfruttando al massimo le potenzialità fornite da Internet, che è poi l'obiettivo di ogni corso on line, al fine di affiancare in maniera stimolante e incitare l'apprendistato di ogni aspirante giocatore.

2 http://it.wikipedia.org/wiki/E-learning

CAPITOLO 1: FASE DI PROGETTAZIONE

1.1 Principali Requisiti Progettuali

La piattaforma Chess-Trainer si prepone di fornire una scacchiera multimediale e dinamica al fine dell'insegnamento del gioco degli scacchi, dove il docente può comunicare con i suoi studenti con molta facilità attraverso l'utilizzo di diversi strumenti progettati ad hoc.

L'insegnamento on-line sfrutta le potenzialità rese disponibili dalla rete per fornire formazione sincrona e/o asincrona agli utenti, che possono accedere ai contenuti dei corsi in qualsiasi momento e in ogni luogo in cui esista una connessione internet.

Chess Trainer è uno strumento di formazione a distanza, una piattaforma di *e-learning*; visto in quest'ottica i criteri principali di progettazione sono:

- *interattività*: necessità di coinvolgere direttamente il discente
- *multimedialità*: effettiva integrazione tra diversi media per favorire una migliore comprensione dei contenuti
- *interazione umana*: con i docenti/tutor per favorire, tramite le tecnologie di comunicazione in rete, la creazione di contesti collettivi di apprendimento.
- *modularità*: ossia la possibilità di organizzare i contenuti di un corso secondo gli obiettivi formativi e le necessità dell'utenza. [3]

3 http://it.wikipedia.org/wiki/E-learning

Al suo interno avremo una sezione dedicata alle partite in corso dove è possibile iscriversi ad un tavolo libero e cominciare una nuova lezione, e un'altra sarà dedicata allo storico delle lezioni, un archivio delle partite salvate e organizzate in moduli.

Fulcro del progetto sarà la tavola da gioco, che non deve essere una semplice scacchiera ma una vera e propria aula virtuale dotata di diversi mezzi di comunicazione.

In qualsiasi momento della partita il docente potrà decidere di cambiare a suo piacimento la disposizione dei pezzi dell'intera scacchiera dando così la possibilità di studiare situazioni di gioco particolari, schemi di attacco o di difesa.

Il sistema verrà fornito di un pennarello che il docente potrà usare per tracciare dei percorsi o dei tracciati, per cerchiare qualsiasi elemento della scacchiera e per scrivere direttamente nell'area di gioco che si trasforma in una vera e propria lavagna virtuale; inoltre avrà anche a disposizione uno strumento *("evidenzia casella")* che permette di colorare qualsiasi casa vuota per sottoporla all'attenzione del discente.

A supporto di queste non può mancare una semplice ma essenziale chat testuale, che sia i docenti che gli alunni potranno usare per comunicare durante la lezione e per aggiungere dei commenti o delle note ad ogni mossa effettuata.

L'applicazione verrà progettata in modo tale da essere utilizzabile dai browser più diffusi: *Internet Explorer*, *Firefox*, *Chrome* e *Safari*.

L'interfaccia grafica sarà chiara e lineare ma allo stesso tempo sarà dinamica e interattiva.

1.2 Architettura

1.2.1 Client to Server

La scelta della tecnologia da utilizzare, al fine di fornire a ogni utente un ambiente dinamico nel quale operare, ricade nel preferire un architettura basata su invio di messaggi asincroni fra *client* e server, gestisti da *AJAX*.
Mentre nei sistemi tradizionali l'utente invia una richiesta al server, e quest'ultimo restituisce un'altra pagina, con *AJAX*, questo meccanismo cambia, infatti ci permette di eseguire richieste asincrone dal nostro *client* web al server, e quest'ultimo può rispondere con funzionalità utili, quali:

- frammenti *HTML*
- script da eseguire sul *client*;
- dati di qualsiasi tipo

AJAX è una tecnica multi-piattaforma utilizzabile su molti sistemi operativi, architetture informatiche e browser web, ed esistono numerose implementazioni *opensource* di librerie e *framework*.

La tecnica Ajax utilizza una combinazione di:

- *HTML* (o *XHTML*) e *CSS* per il markup e lo stile;
- *DOM* (*Document Object Model*) manipolato attraverso un linguaggio *ECMAScript* come *JavaScript* o *Jscript* per mostrare le informazioni ed interagirvi;
- L'oggetto *XMLHttpRequest* per l'interscambio asincrono dei dati tra il browser dell'utente e il web server.

• In genere viene usato *XML* come formato di scambio dei dati, anche se di fatto qualunque formato può essere utilizzato, incluso testo semplice, *HTML* preformattato o *JSON*. Questi file sono solitamente generati dinamicamente da *script* lato *server*.[4]

Come già detto su *AJAX* utilizza l'oggetto *XMLHttpRequest* utilizzato per le prime volte in *Internet Explorer* di *Microsoft* e in seguito adottato da tutti i principali *browser*, esso ci permette di effettuare la richiesta di una risorsa (con *HTTP*) ad un server web in modo indipendente dal *browser*. Nella richiesta è possibile inviare informazioni sotto forma di variabili di tipo *GET* o di tipo *POST* in maniera simile all'invio dati di un *form*.
La richiesta è asincrona, il che significa che non bisogna necessariamente attendere che sia stata ultimata per effettuare altre operazioni, stravolgendo sotto diversi punti di vista il flusso dati tipico di una pagina web.[5]

In *AJAX* è molto diffuso l'uso di *JSON* (*JavaScript Object Notation*), un formato particolarmente adatto allo scambio di messaggi in applicazioni *client-server*. Come si desume dall'acronimo è basato sul linguaggio *JavaScript* che fornisce funzioni e metodi per la sua gestione, che rendono il suo utilizzo molto semplice e intuitivo. Supporta i seguenti tipi di dato:

- booleani
- interi, reali, virgola mobile
- stringhe
- *array*.

4 http://it.wikipedia.org/wiki/AJAX
5 http://javascript.html.it/guide/lezione/2565/descrizione-tecnica-e-teorica/

Ogni utente, durante la fase di gioco esegue periodicamente una chiamata *AJAX* al server, invocando la funzione *"check_board()",* che controlla sia se ci sono state delle modifiche sulla scacchiera (nuove mosse, mosse speciali, promozioni), sia se ci sono modiche che riguardano gli strumenti di comunicazione multimediale (messaggi di chat o effetti grafici).
La funzione ritorna il valore "1" nel caso in cui ci sono state delle modifiche che comportano un aggiornamento automatico della pagina, "0" se non è avvenuto nessun cambiamento.

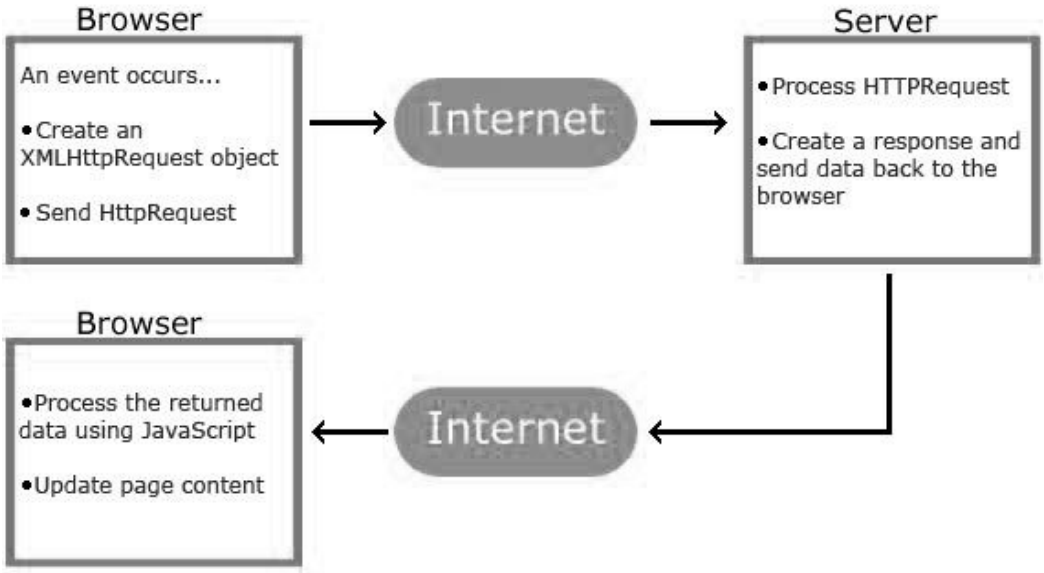

FIGURA 1.1: RICHIESTA AJAX[6]

6 http://www.w3schools.com/php/php_ajax_intro.asp

1.2.2 Progettare un'applicazione WEB con LAMP

Per lo sviluppo dell'applicazione web ho utilizzato la piattaforma *LAMP*, che è considerata la combinazione più diffusa e collaudata negli ambienti di produzione e sviluppo.
LAMP è un 'acronimo che prende il nome dai software da cui è composta, infatti le componenti base di questa tecnologia sono:

- *LINUX*: famiglia di sistemi operativi progettati seguendo le direttive dei sistemi *unix-like*. ovvero sistemi progettati segendo le diretive dei sistemi *UNIX*. Grazie alla portabilità del suo *kernel* sono stati sviluppati sistemi operativi *Linux* per un'ampia gamma di computer, dai cellulari, *tablet* computer e console ai *mainframe* e i *supercomputer*.[7]

- *APACHE* : *Apache HTTP Server*, o più comunemente *Apache* è il nome dato alla piattaforma server Web modulare più diffusa.
  Un web server è un servizio che si occupa di fornire, tramite software dedicato e su richiesta dell'utente (*client*), file di qualsiasi tipo, tra cui pagine web, *RSS* e *XML*. Le informazioni inviate dal server web all'utente viaggiano in rete trasportate dal protocollo *HTTP*.[8]
  Normalmente un server web risiede su sistemi hardware dedicati, ma può essere eseguito su un computer ove risiedano anche altri servizi offerti o su PC utilizzati anche per altri scopi, previa l'installazione del relativo pacchetto software dedicato. [9]

7 http://it.wikipedia.org/wiki/Linux
8 http://it.wikipedia.org/wiki/Apache_HTTP_Server
9 http://it.wikipedia.org/wiki/Web_server

- *MySQL*: è un *Relational Database Management System* (RDBMS), composto da un *client* e un server, entrambi disponibili sia per sistemi *Unix* come *GNU/Linux* che per *Windows*, anche se prevale un suo utilizzo in ambito *Unix*. [10]

  Il modello relazionale è un modello logico di rappresentazione dei dati esso si basa sulla teoria degli insiemi e sulla logica del primo ordine ed è strutturato attorno al concetto di relazione (detta anche tabella). Per il suo trattamento ci si avvale di strumenti quali il calcolo relazionale e l'algebra relazionale.

  Il principio base del modello relazionale è che tutte le informazioni siano rappresentate da valori inseriti in relazioni (tabelle); dunque un database relazionale è un insieme di relazioni contenenti valori e il risultato di qualunque interrogazione (o manipolazione) dei dati può essere rappresentato anch'esso da relazioni (tabelle).

  Il tipo di dato come viene usato nei database relazionali può essere un insieme di numeri interi, un insieme di caratteri alfanumerici, l'insieme delle date, i valori *booleani* vero e falso, e così via. I corrispondenti "nomi di tipo" per esempio saranno le stringhe "*int*", "*char*", "*date*", "*boolean*".[11]

- *Perl*, *PHP* o *Python*: sono i linguaggi di *scripting* che operano lato server nella nostra applicazione, nel nostro caso ci interessiamo *PHP*, che è un linguaggio di *scripting* interpretato, con licenza *opensource* e libera, originariamente concepito per la programmazione Web ovvero la realizzazione di pagine web dinamiche. Attualmente è utilizzato principalmente per sviluppare applicazioni web lato server ma può essere usato anche per implementare script a riga di comando o applicazioni

10 http://it.wikipedia.org/wiki/MySQL
11 http://it.wikipedia.org/wiki/Modello_relazionale

standalone con interfaccia grafica.

Fornisce un'*API* specifica per interagire con *Apache*, nonostante funzioni naturalmente con numerosi altri server web. È anche ottimamente integrato con il database *MySQL*, per il quale possiede più di una *API*. Per questo motivo esiste un'enorme quantità di script e librerie in *PHP*, disponibili liberamente su Internet.

Il *PHP* permette il passaggio di parametri da una pagina all'altra attraverso tre *array* di variabili: *$_GET*, *$_POST* e *$_SESSION*. Il primo tipo di parametro viene passato tramite la stringa che compare nella barra dell'indirizzo del *browser*; il secondo viene passato in background, mentre il terzo rimane persistente durante la sessione.[12]

12 http://it.wikipedia.org/wiki/PHP

1.3 Casi d'uso e diagrammi delle classi

In ingegneria del software, *UML* (Unified Modeling Language) è un linguaggio di modellazione basato sul paradigma *object-oriented*.

UML consente di costruire modelli *object-oriented* per rappresentare domini di diverso genere. Nel contesto dell'ingegneria del software, viene usato soprattutto per descrivere il dominio applicativo di un sistema software e/o il comportamento e la struttura del sistema stesso.[13]

L'utilizzo di una modellazione visuale, come *UML*, implica l'utilizzo di diagrammi che ci aiutano a vedere ed esaminare meglio il quadro generale e le relazioni tra gli elementi del software, e allo stesso tempo ci permette di ignorare e nascondere dettagli poco interessanti. Lavorando in modalità visuale sfruttiamo le capacità del nostro cervello di comprendere rapidamente simboli, unità e relazioni.[14]

I casi d'uso sono requisiti, soprattutto funzionali, che indicano cosa farà il sistema e come si comporteranno gli attori al uso interno.

Un attore è qualcosa o qualcuno dotato di comportamento, anche il sistema stesso può essere considerato un attore. *UML* fornisce una notazione dei diagrammi dei casi d'uso che consente di illustrare gli attori dei casi d'uso e le relazioni tra di essi.

Nella figura 1.2 vengono descritti attori e casi d'uso della applicazione Chess Trainer. Distinguiamo due attori: **Docente** e **Studente**.

Prima di dare inizio alla lezione e cominciare a giocare il docente deve creare una nuova partita; ogni utente studente, dopo aver effettuato il *login* può decidere di entrare nella sezione *Archivio* e guardare un *replay*, o può consultare la lista delle partite libere per scegliere la lezione da seguire.

13 http://it.wikipedia.org/wiki/Unified_Modeling_Language

14 Craig Larman, *Applcare UML e i Pattern – Analisi e Progettazione orientata agli oggetti.* PEARSON 2005. pag 15.

Alla dinamica del caso d'uso *Gioca* sono collegate delle relazioni di tipo *"extend",* rappresentate con una freccia tratteggiata, che indicano delle funzioni rappresentata dal caso d'uso "*estendente*" (alla base della freccia) che possono essere utilizzate dalla funzione "*estesa*" (alla punta della freccia).

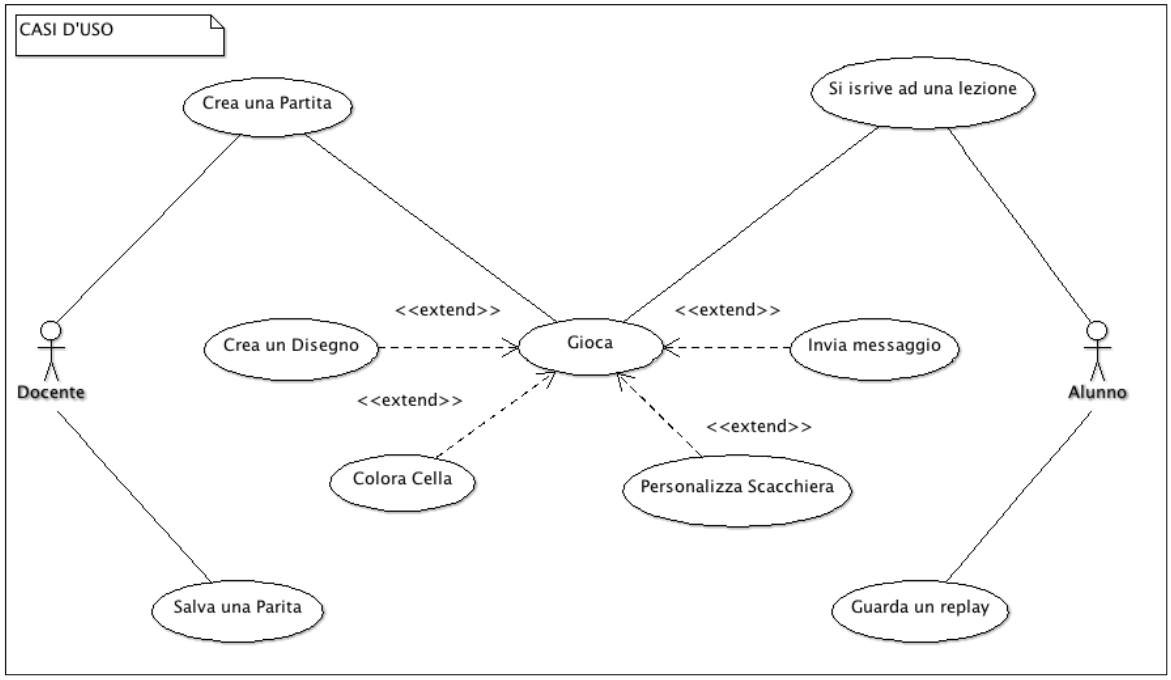

FIGURA 1.2: DIAGRAMMA DEI CASI D'USO

Invece i diagrammi delle classi sono utilizzati da *UML* per illustrare classi, interfacce e relazioni associative. Essi sono utilizzati per la modellazione statica degli oggetti e per la definizione di un modello di dominio. [15]

Il diagramma rappresentato in figura 1.3 descrive varie entità della nostra applicazione (i rettangoli) con le relative relazioni o *link* (le linee), per una corretta analisi dei requisiti funzionali ho utilizzato la tecnica denominata *analisi nome-verbo.*Attraverso questa tecnica si analizza tutta la documentazione disponibile, selezionando nomi e verbi. I nomi sono potenziali candidati a classi e attributi mentre i verbi sono potenziali candidati a responsabilità di classe.

Il diagramma illustra gli oggetti e le loro relazioni dal punto di vista dello studente; elemento fondamentale dell'applicazione è *Game* che ha i seguenti attributi:

- *id*: intero indentificativo della match
- *title*: titolo della lezione
- *player_1*: id del docente
- *player_2*: id dello studente
- *turn*: indica l'id del giocatore che deve effettuare la mossa

e utilizza i metodo:

- *index()*: per visualizzare la liste delle partite disponibili
- *update()*: per aggiornare la suddetta lista
- *submit()*: iscrive un utente alla lezione.

L'entità *Game* è associata da una relazione *"1:1"* con l'oggetto *Board*, che rappresenta la tavola da gioco e contiene tutti gli strumenti multimediali usati per la comunicazione.

15 Craig Larman, op. cit., pag. 261.

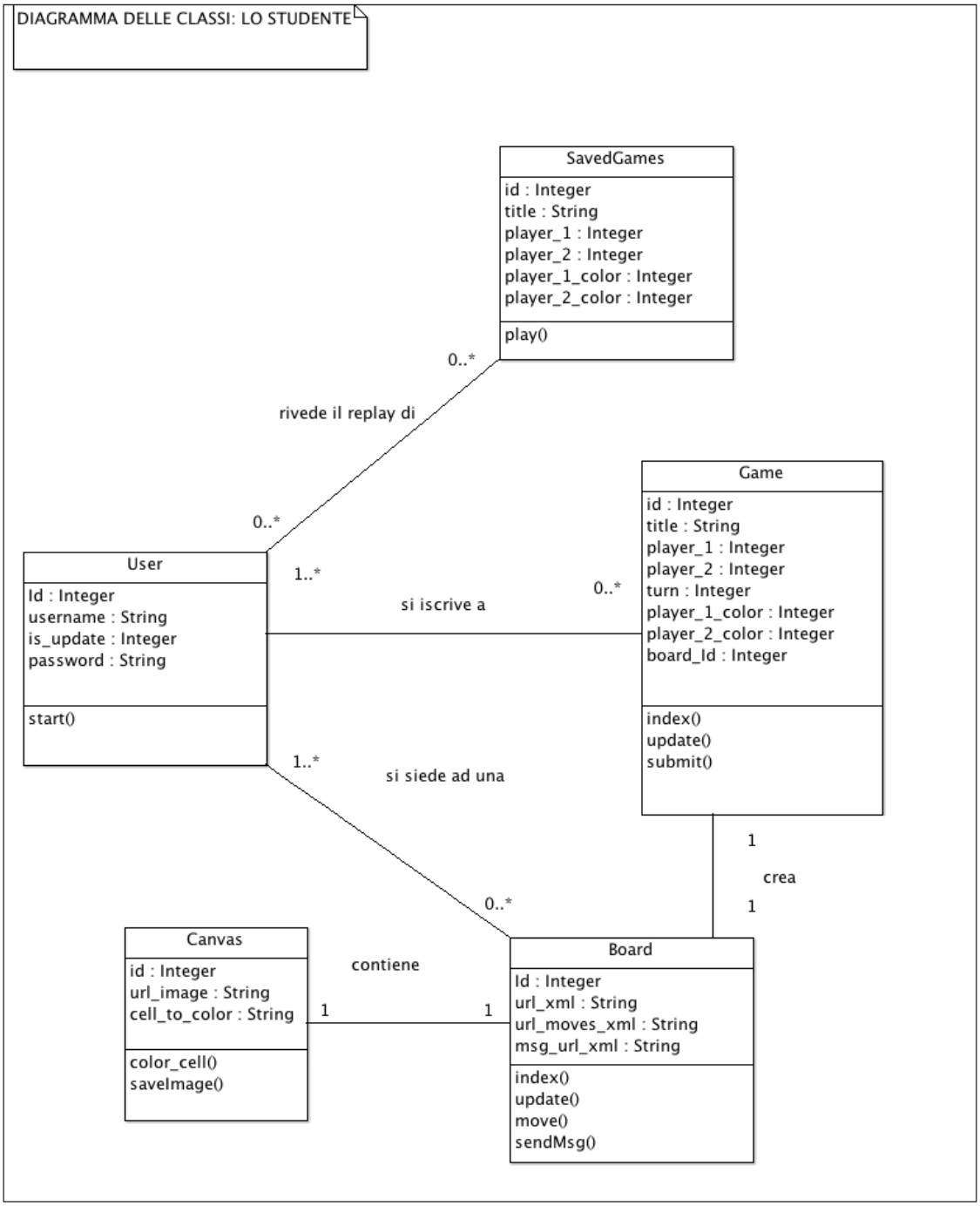

FIGURA 1.3: DIAGRAMMA DELLE CLASSI

1.4 Il pattern MVC E CakePhp

Model-View-Controller (MVC, talvolta tradotto in italiano *Modello-Vista-Controllo*) è un pattern architetturale molto diffuso nello sviluppo di interfacce grafiche di sistemi software object-oriented. Originariamente impiegato dal linguaggio *Smalltalk*, il pattern è stato esplicitamente o implicitamente sposato da numerose tecnologie moderne, come *framework* basati su *PHP* (*Symfony*, *Zend Framework*, *CakePHP*), su *Ruby* (*Ruby on Rails*) e su *Java* (*Swing*, *JSF* e *Struts*).
Il pattern è basato sulla separazione dei compiti fra i componenti software che interpretano tre ruoli principali:

- il *model* fornisce i metodi per accedere ai dati utili all'applicazione;
- la *view* visualizza i dati contenuti nel model e si occupa dell'interazione con utenti e agenti;
- il *controller* riceve i comandi dell'utente e li attua modificando lo stato degli altri due componenti.

Questo schema, fra l'altro, implica anche la tradizionale separazione fra la logica applicativa (in questo contesto spesso chiamata "*logica di business*"), a carico del *controller* e del *model*, e l'interfaccia utente a carico della *view*. [16]
L'intento del pattern MVC è di disaccoppiare il più possibile tra loro le parti dell'applicazione adibite al controllo (*controller*), all'accesso ai dati (*model*) e alla presentazione (*view*).

16 http://it.wikipedia.org/wiki/Model-View-Controller

Questo approccio porta ad innegabili vantaggi come:

- indipendenza tra i business data (*model*) la logica di presentazione (*view*) e quella di controllo (*controller*)
- separazione dei ruoli e delle relative interfacce
- viste diverse per il medesimo *model*
- semplice il supporto per nuove tipologie di *client*: bisogna scrivere la vista ed il *controller* appropriati riutilizzando il *model* esistente. [17]

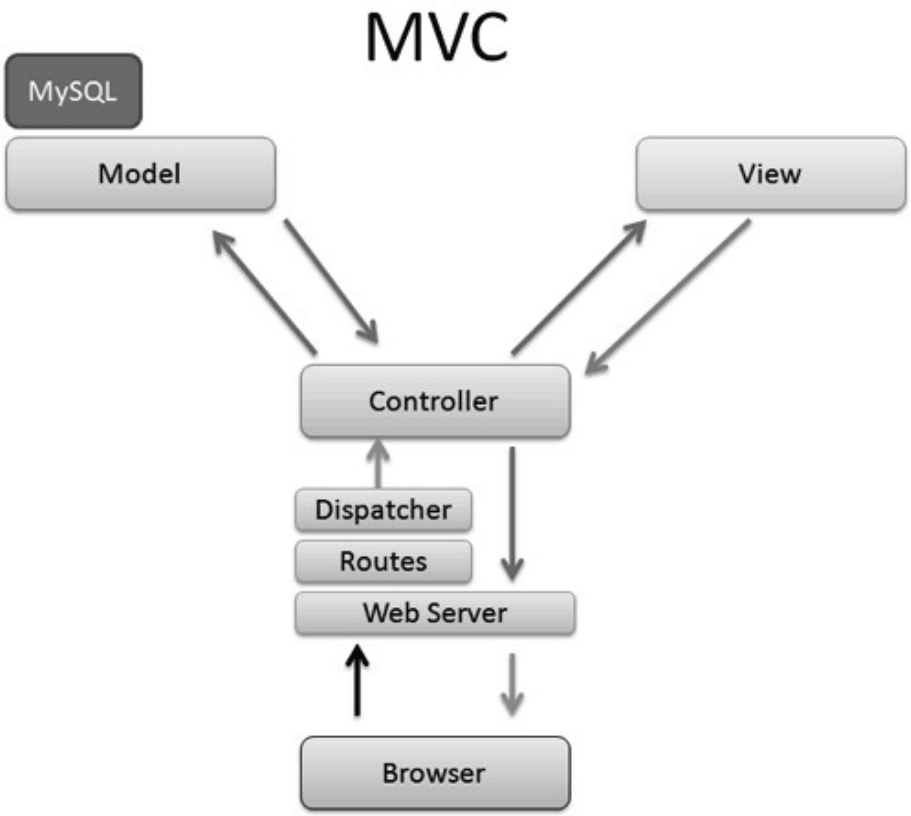

FIGURA 1.4: ESEMPIO DI MVC[18]

17 http://www.mokabyte.it/2003/01/pattern_mvc.htm
18 http://blog.algatux.it/wp-content/uploads/2010/06/mvc.png

CakePhp è un *framework* per il linguaggio *PHP*, ovvero una struttura di supporto su cui un software può essere organizzato e progettato, per la realizzazione di applicazione web. Si ispira al design pattern *MVC* ed è gratuito e *opensource*.
La Figura 1.4 mostra una richiesta *MVC* di base che in *CakePhp* viene gestita nel modo seguente:

1. L'utente studente fa un semplice click sul *link* *http://www.chesstrainer.com/games/sumit* ed il suo *browser* invia una richiesta al *Web Server*;
2. Il *dispatcher* controlla l'*URL* di richiesta (*/games/submit*) e la affida al *Controller* "*GamesController.php"*;
3. Il *controller* esegue la logica specifica dell'azione *submit*, controlla se l'utente abbia effettuato il *login*, controlla se l'aula è libera e infine registra l'utente alla relativa lezione.
   Inoltre, al fine di ottenere l'accesso ai dati dell'applicazione, utilizza i *model* che di solito rappresentano le tabelle del Database e i metodi che fornisce per manipolare i dati.
4. Quando il *controller* ha terminato di effettuare le sue operazioni fornisce i dati da rappresentare nella *view*. In *cakePHP* le *view* sono in formato *HTML*, ma anche *PDF*, *XML* e *JSON* a secondo delle necessità del programmatore. Creata la presentazione della pagina la *view* invia il risultato al browser dello studente.

CakePHP mette a disposizione numerose classi e metodi *built-in*, inclusi nella distribuzione, per gestire le operazioni comuni richieste da una *Web Application*. Vengono divise nelle seguenti categorie:

- *Components*: sono classi che forniscono supporto al *controller*. Tra i componenti più importanti ricordiamo *Session* che fornisce metodi per scrivere, leggere e distruggere una sessione; *Security* e *ACL* per la gestione della sicurezza e per limitare l'accesso a determinati oggetti.
- *Helpers:* sono utili alla creazione di una *view*. Forniscono metodi per formattare il testo e numeri, creare un form *HTML*, inserire codice *JavaScript*.
- *Behavior*: aggiunge funzionalità extra al *model*, che non solo permettono di accedere ai dati sul DB in modo più facilitato ma anche di filtrare i risultati e di effettuare relazioni fra varie tabelle di uno stesso DB.
- *Libreries*: *cakePHP* include alcune librerie di utilità generale che possono essere richiamate ovunque nella applicazione. Sono metodi utili che ci permettono di manipolare facilmente stringhe; di leggere, modificare e salvare file *XML*; per formattare una data.[19]

19 http://book.cakephp.org/2.0

1.4 Due librerie per lo sviluppo dell'interfaccia: JQuery e JQuery UI

"jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is designed to change the way that you write JavaScript."[20]

JQuery è un software *framework JavaScript* - una libreria di funzioni - che ha il fine di semplificare la programmazione lato *client*.

Il core di *JQuery* fornisce metodi che permettono di selezionare elementi di una una pagina *HTML* o per crearne di nuovi. Un elemento può essere selezionato semplicemente usando la sintassi dei selettori *CSS* e grazie a numerosi metodi e funzioni per attraversare e scorrere il *DOM* del documento è possibile risalire a elementi padri, figli, a nodi foglia o ad elementi successivi. La manipolazione del *DOM* avviene per mezzo di metodi che permettono di aggiungere, rimnuovere o sostituire elementi.

Il *framework* fornisce anche metodi per modificare *on-fly* ogni proprietà *CSS* e di arricchire una pagina web effetti grafici come: dissolveze, opacità, effetto sliding.

Nella sviluppo della piattaforma sono state di grande aiuto le funzioni dedicate a gestire le chiamate *AJAX*, tecnologia di scambio di messaggi sulla quale si base tutto il progetto Chess Trainer, rendendo la gestione delle chiamate asincrone estremamente semplice; sono utili per caricare contenuti dinamicamente, per eseguire richieste tipo *GET* o *POST*, per usare oggetti *JSON*.[21]

20 http://jquery.com/
21 http://it.wikipedia.org/wiki/JQuery

"jQuery UI provides abstractions for low-level interaction and animation, advanced effects and high-level, themeable widgets, built on top of the jQuery JavaScript Library, that you can use to build highly interactive web applications."[22]

Con il termine di *JQuery UI* indichiamo un libreria *opensource* che fornisce interazioni ed animazioni, effetti avanzati e *widgets* che possono essere inclusi in una pagina web. E' diviso in quattro aree:

- *Interactions*: comprende tecniche per la creazione di elementi *draggable* e il controllo dell'evento *drop*, avanzate caratteristiche di selezione per liste ed elementi, liste facilmente ordinabili.
- *Widgets*: numerosi *widget* utili per la gestione dell'autocompletamento di *form*, per usare calendari e date, visualizzare contenitori *"Accordion"* e *"Dialog Box"*.
- *Effects*: insiemi di transazioni animate.
- *Utilities*: utilità di basso livello come *"position"*, imposta la posizione relativa di un elemento rispetto ad un altro.

Tutti i *plugins* di *JQuery UI* sono testati per *IE 6.0+*, *Firefox 3+*, *Safari 3.1+*, *Opera 9.6+* e *Google Chrome*.[23]

22 http://jqueryui.com/
23 http://it.wikipedia.org/wiki/JQueryUI

CAPITOLO 2: GLI STRUMENTI DI CHESS TRAINER

2.1 Creare una nuova lezione

Il primo passo da compiere per iniziare una lezione è quello di creare una nuova partita, e questa azione è permessa solo all'utente docente che, dopo aver scelto il nome da assegnare alla sezione di gioco e dopo aver scelto tra il bianco e il nero, preme il pulsante *"Crea"*.
La creazione dell'oggetto *Game* è affidata alla classe *"GamesController"* e ai seguenti metodi:

- *add()*: crea una nuova voce nel Database con tutte le informazioni necessarie, e i file xml relativi alla partita
- *delete()*: cancella la voce dal db e tutti i file xml associati
- *submit()*: permette a qualsiasi studente di iscriversi ad una lezione non occupata
- *index()*: visualizza la lista delle partite libere e di quelle in corso
- *update()*: aggiorna la *view "index.ctp"* ogni volta che viene creato un nuovo match o ogni volta che un utente si iscrive ad una lezione.

Durante la fase di inizializzazione, il *server* genera i seguenti file *XML*:

- *board_IdGame.xml*: contiene tutti i dati necessari per una corretta rappresentazione della tavola da gioco. Il file è finalizzato alla descrizione di tutte le caselle, sia quelle libere che quelle occupate, ed include tutte le informazioni necessarie a disporre i pezzi sulla scacchiera.

- *moves_IdGame.xml*: presenta un lista di tutte le mosse effettuate. In caso di salvataggio del match, il file viene registrato sul *server* e verrà utilizzato nella sezione *Archivio* per il replay delle lezioni.
- *mesgs_IdGame.xml*: contiene tutti i testi dei messaggi inviati durante una lezione, sia quelli inviati dal docente che quelli inviati dall'alunno, e tutte le informazioni relative al messaggio, come la data e l'id dell'utente che scrive.

Qualora la creazione andasse a buon fine, la lista delle partite disponibili viene aggiornata automaticamente. Tutti gli utenti posso accedere a quest'elenco e possono iscriversi alle lezioni, purché queste siano libere.
Infatti se un utente si registra ad una lezione nessun altro potrà accedere nell'aula virtuale, e la riga sarà colorata di rosso, altrimenti per ogni un'aula che risulta ancora disponibile, il *server* fornisce un link - univoco per ogni utente - che permette di iscriversi.
Una volta completata la fase di registrazione, nella *view* dello studente e in quella del docente appare un link che permette di sedersi al tavolo e di cominciare una nuova lezione.

2.2 La Scacchiera

Gli scacchi si giocano su una tavola quadrata, la *scacchiera*, divisa in sessantaquattro caselle, le *case,* organizzate in otto righe, dette *traverse;* le colonne sono otto: mentre le traverse sono numerate da 1 (traversa base dei pezzi bianchi) a 8 (traversa base dei pezzi neri), le colonne sono contraddistinte dalle lettere dell'alfabeto dalla *A* alla *H*. La scacchiera deve essere orientata in modo che la casa nell'angolo in basso a destra di ciascun giocatore sia bianca.[24]
La gestione della scacchiera è affidata al *controller "BoardsController.php"* e in particolare alle funzioni:

- *index()*: si occupa della lettura di *"board_IdGame.xml"*, file univoco per ogni partita. Esso non è altro che un elenco di tutte le caselle vuote ed occupate con le relative informazioni su come disporre i pezzi.
- move(): questo metodo non solo è responsabile di gestire la modifica di *"board_IdGame.xml"* ad ogni mossa, ma ha anche il compito di controllare il meccanismo del cambio turno e di aggiornare il file *"moves_IdGame.xml"*
- *update()*: aggiorna la *view* della scacchiera, è invocato quando viene effettuata una nuova mossa o quando viene postato un nuovo messaggio.

La funzione *"index()"*, dopo aver letto il suddetto file, lo trasforma in un *array* di sessantaquattro elementi; ogni elemento corrisponderà ad una casa della scacchiera, che viene rappresentata come una tabella *HTML*, dove ogni casella è contraddistinta da un valore identificativo, che segue la notazione classica (es: a1,a2,....,a8, b1,b1,.....,b8), e contrassegnata dalla classe *droppable*, attributo che servirà per la gestione dell'evento *drop*.

24 http://it.wikipedia.org/wiki/Scacchi#La_scacchiera

Oltre alla classe *droppable* ogni cella può avere due attributi: o la classe *even,* o la classe *odd*, volti ad assegnare un colore diverso alle celle pari e alle celle dispari, creando così l'effetto a scacchiera, in modo che la casa nell'angolo in basso a destra di ciascun giocatore sia bianca.
Il metodo *"move()"*, come già detto, si occupa di aggiornare il file *"board_IdGame.xml"* e viene chiamato ad ogni nuova mossa. L'evento *drop* è gestito da *jQuery UI Droppable plugin* e viene innescato quando un oggetto di tipo *draggable* viene fatto cadere nella casella.
Oltre a gestire le mosse sulla scacchiera, la funzione *"move()"* ha l'onere di regolare il flusso dei turni della partita, controllando due variabili:

- *game.turn*: questa variabile è contenuta nel record del db che identifica la partita, e il suo valore indica l'id dell'utente che in un dato momento è abilitato ad effettuare una mossa. Il valore assegnato di default è l'id del giocatore di colore bianco, e ogni volta che viene chiamata la funzione *"move()"* il valore viene aggiornato con l'id del giocatore di turno.
- *is_update*: è un campo della tabella *Users* del database; se il valore di questo campo è uguale a "0" allora vuol dire che la vista dell'utente non è aggiornata, "1" altrimenti.

Dopo aver aggiornato il file xml che controlla la scacchiera, e dopo aver provveduto a gestire il flusso dei turni, il metodo aggiunge una nuova mossa alla lista contenuta in *"moves_IdGame.xml"* e infine fa un *redirect* su *boards/index* per ritornare il codice *HTML* aggiornato.

2.3 I pezzi della scacchiera

I pezzi iniziali del gioco sono in totale trentadue, sedici per ciascun colore: un re, una donna (detta anche regina), due alfieri, due cavalli, due torri e otto pedoni; come sappiamo, l'obiettivo del gioco è dare *scacco matto*, ovvero attaccare il re avversario senza che esso abbia la possibilità di sfuggirvi.
In un primo momento, ho pensato di realizzare i pezzi tramite delle immagini in formato png, ma dopo un'attenta ricerca sul web ho constatato, con un po di stupore, l'esistenza di un insieme di caratteri *UNICODE* e *HTML* adibito alla rappresentazione dei pezzi di una scacchiera.
Questa soluzione risulta essere molto efficiente in quanto il *client* browser non deve caricare tutte le immagini per rappresentare gli elementi di una scacchiera, ma gli basta solo convertire il codice come se fossero delle semplici lettere accentate o dei simboli speciali.
Al momento della creazione della scacchiera e ad ogni suo aggiornamento, tenendo in considerazione le variabili *turn* e *user.color* il sistema rende i pezzi *draggable*.
La pagina successiva presenta un'immagine con la codifica *UNICODE* e *HTML* di tutti i pezzi divisi per colore.

Chess Symbols Unicode.org chart (PDF)			
Name	**Symbol**	**Codepoint**	**HTML**
WHITE CHESS KING		U+2654	&#9812;
WHITE CHESS QUEEN		U+2655	&#9813;
WHITE CHESS ROOK		U+2656	&#9814;
WHITE CHESS BISHOP		U+2657	&#9815;
WHITE CHESS KNIGHT		U+2658	&#9816;
WHITE CHESS PAWN		U+2659	&#9817;
BLACK CHESS KING		U+265A	&#9818;
BLACK CHESS QUEEN		U+265B	&#9819;
BLACK CHESS ROOK		U+265C	&#9820;
BLACK CHESS BISHOP		U+265D	&#9821;
BLACK CHESS KNIGHT		U+265E	&#9822;
BLACK CHESS PAWN		U+265F	&#9823;

FIGURA 2.1: CODIFICA PEZZI SCACCHIERA[25]

25 http://en.wikipedia.org/wiki/Chess_symbols_in_Unicode

2.4 Strumenti per la comunicazione

L'intento del lavoro che si vuole presentare con questa tesi è di fornire all'utente una piattaforma web che sia il più possibile efficiente. Dal momento che il suo obbiettivo è l'insegnamento, la pratica e l'affinamento delle tecniche di un gioco, si è dunque riflettuto a lungo anche sull'aspetto comunicativo, educativo, ed anche ludico del progetto. In sostanza: il lavoro svolto deve essere funzionale ed accessibile ai più, e non deve trascurare le esigenze del fruitore. Il sistema con cui l'utente interagisce è una parte essenziale del processo che conduce all'apprendimento; va progettato rispettando sia i principi di usabilità, che quelli relativi all'esperienza d'uso, al fine di rendere l'esperienza dell'utente più piacevole, divertente, coinvolgente e utile. L'interattività dei nuovi media consente una forma più attiva e indipendente di apprendimento: gli studenti sono in grado di manipolare le caratteristiche dello studio (ordine, velocità, contenuti), imparano esplorando e sperimentando in ambienti altamente stimolanti. Tuttavia sarebbe stato un errore, ad esempio, se l'utente collegatosi, si fosse trovato solamente difronte al "Web", ovvero se avesse ricevuto passivamente un quantitativo di informazioni, che in fin dei conti, solo, non avesse saputo poi mettere in pratica.

Utilizzando invece lo strumento interattivo della chat, il praticante si trova davanti una persona, con cui può comunicare, a cui può chiedere informazioni e da cui può ricevere chiarimenti. L'applicazione creata risulta così interattiva, dinamica e semplice da usare, ricca di strumenti adatti a migliorare il processo di apprendimento dello studente. Nei corsi online l'orientamento dominante deve essere proprio la creazione di forme e metodologie di apprendimento ad alto coinvolgimento, sviluppate in un contesto collaborativo, con attività orientate allo sviluppo di progetti che abbiano un riscontro pratico.[26]

26 http://it.wikipedia.org/wiki/Interaction_design_in_ambienti_e-learning

2.4.1 La Chat

Il termine chat (in inglese, letteralmente, *"chiacchierata"*), viene usato per riferirsi a un'ampia gamma di servizi sia telefonici che via Internet.[27]
La chat è uno strumento indispensabile per il corretto svolgimento della lezione, e per questo che viene posta al di sotto della scacchiera per essere facilmente visibile e a portata di mano. Come già detto in precedenza ogni messaggio inviato da qualsiasi utente è memorizzato nel file *"msgs_IdGame.xml"*.
La chat è gestita da *"ChatsController.php"*, che contiene due metodi:

- *send()*: si occupa di inviare il messaggio al *server* e di aggiornare il file xml. Per ogni messaggio viene memorizzato l'utente e la data di invio.
- *update()*: aggiorna l'interfaccia grafica della *chatbox*.

Graficamente la chat lato *client* si presenta come un *form* composto da una *textarea* e da un *input* di tipo testo per l'invio del messaggio.
Ad ogni invio viene innescato l'evento *submit*, che registra il messaggio in una variabile e la invia tramite una chiamata *AJAX* al *server*. Questo evento chiama il metodo *"send()"*, che dopo aver registrato il messaggio, si preoccupa di aggiornare il corrispettivo valore *User.is_update* del destinatario, settandolo uguale a "0", così facendo il destinatario potrà aggiornare la sua *chatbox* e visualizzare il messaggio con il mittente e la data di spedizione.
La chat è dotata anche di una funzionalità particolare, infatti dopo aver effettuato una mossa, qualsiasi utente può associare ad essa una nota o un commento che verrà visualizzato nella fase di replay della partita. In questo caso la chat usa una notazione simile a quella utilizzata nei sistemi *IRC*, ovvero ogni messaggio preceduto dal comando *"/note"* non verrà visualizzato nella *chatbox* ma verrà memorizzato nel tag *note* del file *XML* che contiene la liste delle mosse.

27 http://it.wikipedia.org/wiki/Chat

2.4.2 La scacchiera come lavagna multimediale

Il sistema Chess Trainer è dotato di alcuni strumenti che rendono l'insegnamento più interattivo e piacevole, sono:

- *Il pennarello*: attraverso questo strumento il docente potrà tracciare frecce o linee, disegnare dei percorsi o scrivere direttamente sulla scacchiera, trasformando il tavolo da gioco in una lavagna multimediale.
- *Strumento ""evidenzia casella""*: è un vero e proprio evidenziatore che permette al docente di selezionare delle case da segnalare all'utente.
- *La Gomma*: Cancella ogni elemento grafica dalla *canvas*.

Questi strumenti sono disponibili solo per l'utente docente, e sono accessibili attraverso un pannello di controllo personalizzato. Sono gestiti da un'unica classe *controller "CanvasController.php"* e condividono la stessa tabella *SQL*, ma hanno *view* differenti *"canvas_marker.ctp"* per il pennarello e *"canvas_color_cell.ctp"* per lo strumento evidenziatore, mentre la gomma non ha bisogno di un file *template*, in quanto provvede solamente ad effettuare una chiamata *AJAX* al metodo *"clear()"*, che cancella ogni elemento grafico creato precedentemente.

Le *view "canvas_marker.ctp"* e *"canvas_color_cell.ctp"* sono entrambe visualizzate da *FancyBox,* un *tool* per la presentazione di immagini o di contenuti *HTML*, che offre un metodo piacevole ed elegante per aggiungere effetti grafici alla nostra pagina, implementato dal *framework JQuery*.

Nella *view* del pennarello innanzitutto viene visualizzata la scacchiera aggiornata, e poi un apposito script *JavaScript* crea ed aggiunge un nuovo elemento alla pagina: si tratta del *tag canvas*, che con un piccolo espediente *CSS* viene posizionato al di sopra della scacchiera, dando così l'effetto di avere una lavagna che ha come sfondo la scacchiera aggiornata, sulla quale disegnare.

Canvas è un'estensione dell'*HTML* standard che permette il *rendering* dinamico di immagini bitmap gestibili attraverso un linguaggio di *scripting*, consiste in una regione disegnabile, definita in codice *HTML5* con gli attributi "*height*" e "*width"*.

Il codice *JavaScript* può accedere all'area con un set completo di funzioni per il disegno, permettendo così la generazione dinamica di immagini.

Alcuni usi possibili di *Canvas* includono i grafici, l'animazione e la composizione di immagini. [28]

Il metodo *JavaScript getContext('2d')* preleva il contesto, ovvero un oggetto che offre al programmatore una serie di metodi per operare con la *canvas* nel modo desiderato (nel nostro caso specifico si intende lavorare con le funzioni grafiche bidimensionali, cioè *2D*).[29]

Di seguito elenco i metodi utilizzati per disegnare un'immagine a piacere:

- *beginPath()*: Inizializza un nuovo *path*.
- *moveTO(x,y):* Muove il *path* fino al punto specificato senza creare una linea;
- *lineTo(x,y)*: Crea una linea che parte dell'ultimo punto del *path* fin'ora disegnato fino alle coordinate passate come parametro;
- *closePath()*: Crea una linea dall'ultimo punto del *path* al primo. Completa il *path*.[30]

Per disegnare un'immagine "a mano libera" bisogna gestire ogni *mouse events* all'interno della *canvas*, per far ciò utilizziamo la libreria *JQuery*, che ci fornisce una serie di metodi utili al nostro fine:

28 http://en.wikipedia.org/wiki/Canvas_element
29 http://www.mrwebmaster.it/html/articoli/tag-canvas-html5_1255.html
30 http://www.w3schools.com/html5/html5_ref_canvas.asp

- *mousedown()* e *mouseup()*: rispettivamente verificano quando si preme un tasto del mouse sull'oggetto e quando si rilascia.
- *mousemove()*: si verifica quando il puntatore del mouse si muove all'interno dell'oggetto associato.
- *mouseleave()*: si innesca quando il puntatore esce dall'aria controllata. [31]

Completato il disegno, per salvare l'immagine sul *server* basta premere il bottone "*Salva Immagine"* che chiama attraverso *AJAX* il metodo *"save_image()"*, la particolarità di questa chiamata a dispetto delle altre viste in precedenza è nel paramatro *"data"*, che in questo caso contiene l'immagine disegnata all'interno della *canvas* codificata in *base64*. Il file viene creato della stessa dimensione della scacchiera e in formato *PNG*, per essere poi sovrapposto alla tavola da gioco. Per ogni immagine creata viene aggiunta una voce alla tabella *Canvas* del BD dove viene indicato l'indirizzo del *server* dove è salvata l'immagine. L'implementazione del *tool "*"*evidenzia casella*"*"* è meno difficoltosa del precedente strumento, difatti non necessita dell'elemento *canvas*, ma sfrutta le potenzialità di *JQuery* di modificare facilmente e *on-fly* il *DOM* della pagina html unita ad una perfetta gestione del mouse events *click*.
Infatti, in modalità *"evidenzia casella"*, ad ogni click su una casella vuota viene aggiornato un stringa contente gli id delle case da evidenziare, cliccando sull'oggetto *colora caselle*, viene creato un record nel DB relativo indicando nel campo adeguato la stringa con gli id da colorare.
Al momento di aggiornare la pagina, il *client* riceverà la lista degli ID da colorare, dunque l'effetto visivo verrà creato modificando l'opacità di ogni oggetto con il comando *CSS3 opacity.*

31 http://www.w3schools.com/tags/ref_eventattributes.asp

2.5 Edit Board

"Edit Board" è un *tool* fondamentale per lo svolgimento della lezione dal momento che ha svariate funzionalità che ci permettono sia di gestire dei casi particolari, sia di personalizzare la scacchiera a nostro piacimento; inoltre è disponibile sia per il docente che per l'alunno. La sua icona ha la forma di una piccola scacchiera ed visibile nei rispettivi pannelli di controllo.
Nel gioco degli scacchi vi sono dei movimenti o dei comportamenti non convenzionali dei pezzi, come:

- *Arrocco*: è una mossa particolare che coinvolge il re e una delle due torri. È l'unica mossa che permette di muovere due pezzi contemporaneamente. Consiste nel muovere il re di due case a destra o a sinistra in direzione di una delle due torri e successivamente muovere la torre (verso la quale il re si è mosso) nella casa accanto alla casa occupata dal re. Può avvenire solo in condizioni particolari.[32]
- *En passant*: È una mossa che coinvolge esclusivamente i pedoni e può essere eseguita da ciascun pedone che si trovi nelle condizioni adatte. L'en passant è legato alla caratteristica del pedone di spostarsi di due case quando abbandona la sua posizione di partenza. Quando un pedone, muovendosi di due passi (quindi per la prima volta), finisce esattamente accanto (sulla stessa traversa e su colonna adiacente) ad un pedone avversario, quest'ultimo può catturarlo come se si fosse mosso di un passo solo. I diagrammi illustrano le tre fasi di questa manovra.[33]

32 http://it.wikipedia.org/wiki/Arrocco
33 http://it.wikipedia.org/wiki/En_passant

- *Promozione*: se un pedone bianco raggiunge l'ottava traversa o un pedone nero la prima può essre promosso al rango di pezzo dello stesso colore. Tale pezzo viene scelto dal giocatore tra donna, torre, alfiere e cavallo indipendentemente da quali e quanti pezzi siano già presenti sulla scacchiera.[34]

Si ricordi che il sistema, al fine di garantire un corretto svolgimento della partita, deve poter gestire queste situazioni particolari. A tale fine, una volta attivata la schermata *"Edit Board",* è possibile compiere una serie di azioni quali:

- cancellare un qualsiasi pezzo
- cambiare qualsiasi pedone in un rango diverso
- aggiungere un nuovo pezzo
- muovere qualsiasi pezzo senza nessuna restrizione.

Usando e combinando questi comandi è possibile gestire i casi sopra descritti, quali l'arrocco, l'en passant e la promozione.
"Edit Board", oltre che le funzionalità appena esplicate, può essere impiegato per altri fini, come la personalizzazione della schacchiera al fine di creare situazioni particolari di gioco. Come già descritto, la piattaforma Chess Trainer permette in qualunque momento, sia all'inizio della partita che durante il match, di spostare qualunque pezzo della scacchiera: di conseguenza la lezione in qualsiasi momento può sviluppare una serie di quesiti da porre all'attenzione dello studente, mostrando particolari strategie di attacco o di difesa.

34 http://it.wikipedia.org/wiki/Promozione_(scacchi)

2.6 L'archivio delle lezioni

Fornire dei contenuti digitali utili in qualsiasi momento e in ogni luogo in cui esista una connessione internet è una caratteristica fondamentale di ogni piattaforma *e-learning*.

Chess Trainer dedica un'ampia sezione per la consultazione delle lezioni salvate, raggruppate in un archivio facilmente consultabile da ogni utente registrato. Il replay della partita mostra ogni fase di gioco dall'inizio della partita fino alla conclusione della lezione, mostrando anche ogni eventuale commento pubblicato durante il match da uno dei due giocatori.

In conclusione ogni studente ha a disposizione un potente strumento che lo aiuta a studiare e approfondire la sua conoscenza e gli permette l'autogestione e l'autodeterminazione del proprio apprendimento.

Mentre il docente potrà usare questo strumento per creare delle lezioni o del materiale didattico e organizzarlo in moduli, adempiendo così al requisito fondamentale dell'*e-learning* della *modularità*.

Per salvare una partita il docente deve cliccare sull'icona a forma di dischetto nel suo pannello di controllo, questo bottone effettua una chiamata *AJAX* al metodo *"save_game()"* definito nella classe *SavedGames*.

All'interno del *server* il salvataggio avviene con la creazione di un nuovo file che consiste in una "copia carbone" del file *"moves_GameID.xml"* e nella aggiunta di un record nella tabella *mysql saved_games.*

L'utilizzo del linguaggio XML consente una migliore atomizzazione dei contenuti e una più efficace esportabilità sui diversi supporti.[35]

35 http://it.wikipedia.org/wiki/E-learning

CONCLUSIONE

Chess Trainer è un valido strumento capace di offrire una piattaforma di *e-learning* completa, esso infatti soddisfa tutti i requisiti funzionali tipici di ogni sistema telematico che si rispetti. I principi progettuali della piattaforma - finalizzata all'insegnamento a distanza del gioco degli scacchi - si focalizzano nel facilitare l'interattività e la dinamicità del processo di insegnamento che viene affiancato da numerosi e potenti strumenti di comunicazione sia sincroni che asincroni.

Infatti il sistema oltre a fornire una tavola da gioco dinamica e semplice da usare, integra una lavagna multimediale che permette di disegnare "a mano libero" direttamente sulla scacchiera e di colorare caselle di gioco con colori diversi; inoltre è possibile anche salvare sul server intere lezioni e rivederle in qualunque momento.

Per lo sviluppo di questa tesi di laurea sono state impiegate e integrate fra di loro le tecnologie più popolari e più utilizzate nel campo della programmazione di applicazioni web.

Una di queste è la piattaforma LAMP (Linux+Apache+MySQL+PHP) che a sua volta è una combinazione di altre tecnologie che insieme forniscono un potente web server, ospitato su un sistema operativo unix-like, e un database di tipo relazionale per la rappresentazione dei dati più significativi.

Nelle prime fasi di progettazione, dopo un'attenta analisi dei requisiti fondamentali, ho redatto una serie di digrammi UML che indicano cosa farà il sistema e come si comporteranno gli attori al uso interno.

Lo schema di progettazione seguito si basa sulla separazione dei compiti attraverso il pattern architetturale *Model-View-Controller* (abbreviato spesso in

MVC), che consiste nel scissione delle componenti software che implementano il modello delle funzionalità di business (*model*), dai componenti che implementano la logica di presentazione (*view*) e da quelli che ricevono i comandi dell'utente e li attua modificando lo stato degli altri due componenti.[36]

Il tipo di comunicazione scelto per permettere a client browser e server di interagire è asincrono, ovvero il mittente spedisce il messaggio e continua ad effettuare le proprie operazioni (AJAX), questa tipologia di messaggi permette all'applicazione di aggiornare automaticamente una pagina senza un intervento manuale dell'utente.

La progettazione lato client utilizza il framework JQuery, libreria di funzioni Javascript che semplificano il lavoro del programmatore, infatti con poche righe di codice è possibile effettuare svariate operazioni, come ad esempio ottenere l'altezza di un elemento o farlo scomparire con effetto dissolvenza; altresì JQuery include delle funzionalità per la gestione di chiamate asincrone AJAX con molta semplicità e flessibilità.

L'utilizzo di questo framework, unito anche alle scelte progettuali descritte in questo documento, garantisce che l'applicazione sia supportata dai maggior browser in uso, infatti tutti i *plugins* di *JQuery* sono testati per *IE 6.0+*, *Firefox 3+*, *Safari 3.1+*, *Opera 9.6+* e *Google Chrome.*

36 http://it.wikipedia.org/wiki/Design_pattern#Pattern_architetturali

SITOGRAFIA

http://www.apache.org/

http://aptana.com/

http://book.cakephp.org/2.0/en/

http://www.eclipse.org/

http://en.wikipedia.org/wiki/Chess_symbols_in_Unicode

http://forum.html.it/forum/

https://help.ubuntu.com/community/ApacheMySQLPHP

http://it.wikipedia.org

http://jquery.com/

http://jqueryui.com/

http://www.lamphowto.com/

http://php.net/manual/en/index.php

http://stackoverflow.com

http://www.w3schools.com/html/default.asp

http://www.w3schools.com/css/default.asp

http://www.w3schools.com/js/default.asp

BIBLIOGRAFIA

J. Chaffer K. Swedberg - *Learning jQuery, Third Edition*. PAPERBACK 2011.

C. Larman - *Applicare UML e i Pattern Analisi e Progettazione orientata agli oggetti*. PERSON 2005.

Murphy & Clark & Studholme - *Beginning HTML5 and CSS3: Next Generation Web Standards*. APRESS, 2010

R. Nixon - *Learning PHP, MySQL, and JavaScript*. O'REILLY MEDIA, 2009.

E. William & D.Lane - *Applicazioni web database con PHP e MySQL*. Tecniche Nuove, Milano, 2005.