

Progetto d'esame per il corso di Laboratorio interdisciplinare B A.A. 2019-2020

Pietro Colombo

pietro.colombo@uninusbria.it

Dipartimento di Scienze Teoriche e Applicate
Università dell'Insubria,

Via O. Rossi, 9 - 21100 Varese – Italy



Il Paroliere (IP)

- Una piattaforma di gioco ispirata al celebre gioco da tavolo
- Elementi del gioco:
 - 16 dadi, riportanti su ogni faccia, una lettera dell'alfabeto;
 - una griglia di dimensione 4x4 sulla quale i dadi, una volta lanciati, verranno disposti;
 - un dizionario, usato per verificare l'esistenza delle parole;
 - una clessidra della durata di 3 minuti.
- Ad una partita de *Il Paroliere* possono partecipare da 2 a 6 giocatori
- Una partita è caratterizzata da un numero variabile di sessioni di gioco.

		Faccia					
		1	2	3	4	5	6
Dado	1	B	A	O	O	Qu	M
	2	U	T	E	S	L	P
	3	I	G	E	N	V	T
	4	O	U	L	I	E	R
	5	A	C	E	S	L	R
	6	R	A	T	I	B	L
	7	S	M	I	R	O	A
	8	I	S	E	E	F	H
	9	S	O	T	E	N	D
	10	A	I	C	O	F	R
	11	V	N	Z	D	A	E
	12	I	E	A	T	A	O
	13	O	T	U	C	E	N
	14	N	O	L	G	U	E
	15	D	C	M	P	A	E
	16	E	R	I	N	S	H

Il gioco

- Lo scopo del gioco è di misurare l'abilità dei giocatori nel riconoscere parole componibili con le lettere mostrate dai dadi.
- Una sessione di gioco inizia con il lancio e la disposizione casuale dei 16 dadi sulla griglia 4x4.

G	A	U	T
P	R	M	R
D	O	L	A
E	S	I	C

Esempio di griglia



I dadi che l'hanno
generata



3	1	4	2
15	5	7	6
11	9	14	12
16	8	10	13

Il gioco

- Una volta disposti i dadi viene girata la clessidra.
- L'obiettivo per i giocatori, è di annotare, entro 3 minuti (la durata della clessidra), il maggior numero di parole, di lunghezza maggiore possibile, composte concatenando le lettere indicate nella griglia.
 - Le parole vengono composte partendo dalla lettera presente in una cella della griglia, e visitando le celle adiacenti
 - È possibile muoversi verticalmente, orizzontalmente od obliquamente di una posizione in qualsiasi direzione, evitando di visitare la stessa cella più di una volta.
 - Le parole considerate ammissibili sono quelle indicizzate nel dizionario, ovvero parole corrispondenti ad un vocabolo utilizzabile come chiave di ricerca:
 - sostantivi o aggettivi nella forma singolare
 - verbi all'infinito

Il gioco

- Esempio di parole ammissibili:
 - DOSE, ORMA, TRAMA, PRODE ed ARDESIA
- Esempio di parole non ammissibili:
 - SILOS, TRAUMA, MORDE

G	A	U	T
P	R	M	R
D	O	L	A
E	S	I	C

Il gioco

- Al termine dei 3 minuti caratterizzanti la sessione di gioco, ogni giocatore comunica la lista di parole identificate
- Vengono scartate:
 - Le parole non ammissibili
 - Le parole reputate ammissibili che sono state proposte da più giocatori
- Per ogni parola rimanente, un giocatore riceve un quantitativo di punti:
 - per ogni parola di 3 o 4 lettere: 1 punto
 - per ogni parola di 5 lettere: 2 punti
 - per ogni parola di 6 lettere: 3 punti
 - per ogni parola di 7 lettere: 5 punti
 - per ogni parola di 8 o più lettere: 11 punti

Il gioco

- Vince la partita il giocatore che raggiunge per primo la soglia di 50 punti guadagnati.
- Se in una singola sessione di gioco nessun giocatore riesce ad accumulare 50 punti, si tiene traccia dei punti guadagnati da ogni giocatore, e il gioco viene reiterato ripartendo con una nuova sessione.
- Le reiterazioni terminano quando il punteggio complessivo di un giocatore raggiunge la soglia.

Requisiti

- Realizzare una piattaforma di gioco ispirata al gioco in scatola.
- Supporta utenti con 2 profili distinti:
 - il *concorrente* di gioco, che organizza e partecipa a delle partite;
 - l'*amministratore*, che gestisce le componenti di backend della piattaforma.

Applicazione client/server

- La piattaforma IP consiste di:
 - *serverIP*, un modulo server che interfacciandosi con un DBMS relazionale (PostgreSQL), e con un dizionario della lingua italiana, fornisce servizi di back-end;
 - *playerIP*, un modulo client che fornisce servizi designati per i giocatori.
- IP dovrà essere realizzata in modo tale da supportare l'interazione in parallelo con più utenti connessi alla piattaforma da postazioni differenti.
- IP dovrà essere realizzata in Java utilizzando tecnologie che permettano un'implementazione efficiente dei servizi.

Requisiti

- Servizi riguardanti:
 - gestione degli utenti
 - gestione e gioco delle partite
 - monitoraggio del sistema
- La realizzazione di tali servizi richiede:
 - suddivisione delle funzionalità suportate da *playerIP* e *serverIP*
 - definizione di opportuni protocolli di interazione tra queste componenti

Avvio del server

- All'invocazione di *serverIP* deve essere richiesto di specificare:
 - le credenziali per accedere a *dbIP*, un database di supporto all'esecuzione dei servizi di *IP*,
 - l'host del database.
- Se *dbIP* non contiene un profilo di utente amministratore:
 - *serverIP* deve avviare un processo di registrazione, richiedendo all'utente amministratore che ha invocato l'esecuzione un identificativo *uid* e una password *pwd*.
- Le credenziali registrate (*uid* e *pwd*) potranno essere utilizzate per effettuare l'autenticazione dell'utente amministratore.
 - Al termine della registrazione, o nel caso in cui in *dbIP* sia già presente un profilo di amministratore, *serverIP* rimane in attesa di richieste di connessione da parte di *playerIP*.

Registrazione utenti

- Un utente che intende accedere a *IP* deve necessariamente registrarsi.
 - *playerIP* gestisce la registrazione di concorrenti
- Il processo di registrazione richiede all'utente di fornire:
 - il nome, il cognome, un nickname, un indirizzo di posta elettronica e una password.
- *L'indirizzo di posta elettronica e il nickname non devono essere già stati usati da altri utenti.*
- Al termine della registrazione la piattaforma invierà all'utente un codice di attivazione da immettere per completare la registrazione.
 - Se l'utente non completa la registrazione entro 10 minuti dalla spedizione del codice il profilo dovrà essere cancellato.

Autenticazione

- Richiesta per l'accesso ad ogni funzionalità offerta dalla piattaforma tramite *playerIP*
 - *ad eccezione del servizio di registrazione utenti e di ripristino password dimenticata*
- Effettuata mediante *playerIP*, specificando:
 - l'indirizzo di posta registrato
 - la password associata al profilo.

Utenti autenticati

- Un utente autenticato può:
 - organizzare una nuova partita,
 - visualizzare le partite organizzate per la quale non si sono ancora chiuse le iscrizioni
 - visualizzare le partite in corso di svolgimento,
 - richiedere la partecipazione, ad una partita organizzata da altri utenti per la quale non sono ancora state chiuse le iscrizioni ed iniziare a giocare una volta terminate le iscrizioni
 - abbandonare una partita per la quale non sono ancora state chiuse le iscrizioni, oppure che è in corso di svolgimento
 - modificare i dati del proprio profilo
 - analizzare le statistiche di utilizzo della piattaforma di gioco

Organizzare una partita

- Un utente autenticato, mediante *playerIP* può organizzare una nuova partita stabilendone il nome, e il numero di giocatori che possono parteciparvi (da 2 a 6).
- L'utente organizzatore viene automaticamente iscritto alla partita organizzata

Visualizzare la lista delle partite

- Un utente autenticato, mediante *playerIP* può visualizzare:
 - l'elenco delle partite organizzate con iscrizioni dei concorrenti ancora aperte
 - l'elenco delle partite già in corso di svolgimento.
- Per ogni partita in elenco viene mostrato il nome, la data, e l'ora di creazione, il numero di giocatori previsti, il numero di quelli iscritti ed i rispettivi nickname.

Richiesta di partecipazione ad una partita

- Un utente autenticato, mediante *playerIP* può richiedere di partecipare ad una partita organizzata da altri utenti con iscrizioni ancora aperte.
- *IP* deve autorizzare la partecipazione se il numero di giocatori già iscritti è inferiore al numero di giocatori previsti per la partita dall'organizzatore.

Svolgimento di una partita

- Al raggiungimento del numero di giocatori necessari a far iniziare una partita, *IP* chiude le iscrizioni e notifica l'inizio imminente del gioco ai giocatori registrati.
- I moduli *player/IP* dei giocatori iscritti visualizzano un timer che effettua un conto alla rovescia di 30 secondi, corrispondente al tempo mancante all'inizio effettivo del gioco
 - i timer dei giocatori devono essere sincronizzati.
- Allo scadere del timer inizia l'effettiva sessione di gioco.

Svolgimento di una partita - avvio sessione

- All'inizio della sessione *IP* simula:
 - il lancio dei 16 dadi, generando casualmente la lettera mostrata da ogni dado, e la disposizione casuale dei dadi nella griglia
 - il ribaltamento della clessidra, facendo partire un timer che effettua un conto alla rovescia di 3 minuti.
- Il modulo *playerIP* dei giocatori dovrà visualizzare:
 - Il nome della partita in corso di svolgimento, e l'ordinale della sessione di gioco avviata
 - la griglia 4x4 riportante le lettere selezionate
 - il timer che indica i minuti e secondi mancanti al completamento della sessione di gioco
 - i nomi e i punteggi di tutti i giocatori partecipanti

Svolgimento di una partita

- *playerIP* deve permettere al giocatore di annotare le parole che ha identificato nella sessione di gioco
 - le parole annotate dagli altri giocatori non sono visibili
- Allo scadere del timer, *IP* dovrà analizzare le parole annotate dai vari giocatori, verificando che ogni parola:
 - sia componibile visitando la griglia in base ai criteri considerati,
 - sia di lunghezza almeno pari a 3 lettere,
 - sia un vocabolo presente nel dizionario,
 - non sia stata proposta da più giocatori.

Svolgimento di una partita

- *playerIP* dovrà mostrare ad ogni giocatore:
 - la lista delle parole identificate da ogni concorrente,
 - il punteggio apportato da ogni parola,
 - Usando i criteri precedentemente illustrati
 - il punteggio acquisito nella sessione di gioco da ogni giocatore,
 - il punteggio cumulativo dell'intera partita per ogni giocatore.
- Nel caso di parole con punteggio associato pari a 0, dovrà essere indicato se la parola:
 - non è derivabile dalla griglia,
 - non è presente nel dizionario,
 - è di lunghezza insufficiente,
 - è stata indicata da altri,
 - oppure una combinazione di questi criteri.

Svolgimento di una partita – verifica parole

- Il giocatore dovrà poter verificare la definizione di qualsiasi parola:
 - nella lista delle parole trovate dagli altri giocatori o nella propria lista.
- *playerIP* dovrà
 - permettere al giocatore di selezionare le parole che intende verificare,
 - mostrare la definizione derivata dal dizionario delle parole selezionate.
- Ogni giocatore avrà fino a 3 minuti per la verifica. *playerIP* dovrà mostrare:
 - un timer che indica il tempo rimanente,
 - un pulsante che dovrà permettere al giocatore di notificare ad *IP* di aver terminato la verifica.
- Il turno di analisi termina quando tutti i giocatori iscritti hanno notificato di aver completato l'analisi, oppure, se tale condizione non si verifica, alla scadenza del timer.

Svolgimento di una partita

- Una volta completata l'analisi, se il punteggio cumulativo di almeno un giocatore raggiunge la soglia di 50 punti, *playerIP* dovrà mostrare il vincitore della partita.
- In caso contrario, *IP* dovrà riproporre una nuova sessione di gioco.

Abbandono di una partita

- *playerIP* deve permettere ai giocatori di abbandonare la partita a cui stanno partecipando.
- Se un giocatore abbandona la partita in corso di svolgimento:
 - la partita viene annullata,
 - tutti i giocatori ricevono notifica dell'annullamento,
 - la partita viene rimossa dalla lista delle partite organizzate.
- Se l'abbandono avviene prima della chiusura delle iscrizioni:
 - il giocatore viene rimosso dalla lista dei concorrenti iscritti;
 - se in seguito all'abbandono la lista dei concorrenti iscritti è vuota:
 - la partita viene annullata e rimossa dalla lista delle partite organizzate.
- *IP* dovrà tener traccia della partecipazione ad ogni sessione di gioco, delle parole identificate e dei relativi punteggi acquisiti.

Modifica dei dati di profilo:

- Un utente, mediante *playerIP* deve poter modificare i dati del proprio profilo: correggendo nome e cognome, cambiando nickname, o password.
- L'impostazione di una nuova password è soggetta all'inserimento della vecchia password.
 - IP dovrà inviare una mail all'indirizzo di posta registrato informandolo del cambio avvenuto (senza mostrare le password).

Reset di password

- *playerIP* deve fornire un servizio per reimpostare una password dimenticata, accessibile anche da utenti non autenticati.
- Il servizio deve:
 - richiedere all'utente di specificare l'email associata al profilo,
 - rimpiazzare la password del profilo con una nuova generata automaticamente,
 - inoltrarla all'indirizzo di posta elettronica associato al profilo.

Monitoraggio

- *playerIP* deve offrire servizi di analisi:
 - Mostrare il giocatore:
 - che detiene il primato di punteggio raggiunto per sessione e per partita, indicando i rispettivi punteggi;
 - che ha giocato più sessioni di gioco in assoluto, indicandone il numero;
 - con la media più alta di punti acquisiti per sessione di gioco e per partita, indicando i rispettivi punteggi;
 - che ha proposto il più alto numero di parole duplicate, ovvero indicate da almeno un altro giocatore nella stessa sessione di gioco, indicandone il numero;
 - che ha proposto il più alto numero di parole che non possono essere derivate dalla configurazione dei dadi mostrata sulla griglia, oppure che non sono presenti nel dizionario, indicandone il numero;

Monitoraggio

- Mostrare la classifica delle occorrenze delle parole valide che sono state correttamente identificate.
- Mostrare le parole la cui identificazione ha causato il più alto incremento di punti, ed indicare l'incremento corrispondente, riferendo, per ognuna, la partita in cui è stata proposta
- Mostrare il numero medio di turni di gioco che caratterizzano una partita interamente giocata, da gruppi di 2, 3, 4, 5 e 6 giocatori
- Mostrare il numero minimo e massimo di turni che caratterizzano una partita interamente giocata da gruppi di 2, 3, 4, 5 e 6 giocatori
- Derivare l'occorrenza media delle lettere mostrate apparse sulla griglia nei vari turni di gioco
- Mostrare le parole di cui è stata richiesta almeno una volta la definizione, ordinandole dalla più richiesta alla meno richiesta.
- Mostrare l'identificativo delle partite in cui è stata richiesta la definizione di una parola data.

Requisiti dati

- *server/IP* fornisce servizi di supporto alla memorizzazione ed analisi dei dati dell'intera piattaforma di gioco.
- Progettare la base di dati che *server/IP* utilizzerà per gestire informazioni relative agli utenti, e alle partite giocate. In particolare, sono di interesse:
 - il profilo dell'amministratore, e dei giocatori regolarmente registrati, caratterizzato dal cognome, il nome, un nickname, un indirizzo email, e una password.
 - lo storico dei turni di gioco e delle partite che sono state giocate e che si stanno giocando, tenendo traccia: delle lettere mostrate sulla griglia, delle parole identificate dai giocatori, dei punti acquisiti con ogni parola, e delle richieste di definizione di parole.

Svolgimento del progetto

- L'obiettivo del progetto consiste nella
 - *progettazione e sviluppo* di un sistema software e della relativa base di dati che soddisfi i requisiti proposti, utilizzando un processo di sviluppo object oriented,
 - produzione di prodotti intermedi (diagrammi UML e gli schemi ER) che descrivano quanto realizzato nelle varie fasi del processo, in conformità alle metodologie seguite a lezione.
- Le attività di analisi e progettazione devono essere adeguatamente documentate facendo uso del linguaggio UML per l'applicazione software e del modello Entity-Relationship (ER) per il database.
- Fare riferimento al documento allegato per vedere come organizzare la presentazione dei prodotti intermedi nella relazione

Svolgimento del progetto

- È richiesto di :
 - progettare l'applicazione avvalendosi dove possibile dell'uso di design patterns
 - realizzare l'applicazione con un'opportuna interfaccia grafica, usando il linguaggio di programmazione Java,
 - progettare e realizzare un database utilizzando PostgreSQL per la sua implementazione (<http://www.postgresql.org>) e JDBC per l'accesso alla base di dati da programma Java (<http://jdbc.postgresql.org/download.html>)
- I servizi dell'applicazione vengono erogati in parallelo a più utenti, e che possono verificarsi accessi concorrenti a risorse condivise.
 - L'applicazione deve quindi essere definita in modo tale da gestire opportunamente l'accesso concorrente alle risorse accedute.

Realizzazione del database

- Attività ed artefatti da produrre nelle fasi di realizzazione:
 - Si ristrutturi, se necessario, secondo le metodologie di progettazione i requisiti descritti. Si scelgano le metodologie per la costruzione dello schema ER, motivando le scelte fatte.
 - Si definisca lo schema concettuale ER per il database, evidenziando le entità e le associazioni di interesse, nonché i vincoli di cardinalità e di identificazione, motivando le scelte effettuate. Altri eventuali vincoli devono essere espressi in linguaggio naturale.
 - Si effettui la ristrutturazione dello schema ER motivando le scelte effettuate.

E' richiesto di produrre un documento di analisi dei requisiti ristrutturato e documentazione associata allo schema ER (ristrutturato e non), con eventuale specifica di vincoli in linguaggio naturale.

Realizzazione del database

- Si effettui la traduzione dello schema ER ristrutturato in un equivalente schema relazionale.
- *E' richiesto di produrre la documentazione associata allo schema relazionale derivato dallo schema concettuale.*
 - Si realizzi il database utilizzando PostgreSQL, e SQL per la definizione dei dati, l'implementazione dei vincoli identificati, e la manipolazione dei dati, secondo le operazioni previste dall'applicazione.
- *Documentare gli script SQL necessari alla creazione della base di dati e dei vincoli definiti sui dati e le query SQL a supporto dei servizi erogati da IP.*

Codifica

- Codice sorgente dell'applicazione dovrà essere definito in modo da facilitarne la comprensione e la manutenibilità:
 - Ogni volta che si identificano due estratti di codice sufficientemente simili, il codice deve essere reingegnerizzato, e il codice duplicato eliminato portando a fattore comune le funzionalità condivise.
 - Si consiglia l'utilizzo di strumenti di refactoring forniti da ambienti di sviluppo quali, ad esempio, Eclipse / NetBeans.
 - E' opportuno considerare che esistono delle "regole di stile" per ogni linguaggio di programmazione, quindi per indentazione, nomi, parentesi, commenti, etc.

Documentazione del codice

- Generare documentazione javadoc documentando i metodi e gli attributi con visibilità *public* o *protected* di ogni classe con visibilità *public*.
 - spiegare come usare una determinata classe o interfaccia, non come questa sia stata effettivamente implementata
- Prima di commentare il codice perché altrimenti incomprensibile, tentare le seguenti strade:
 - Se si sono usati nomi poco chiari, sostituirli con nomi che spieghino lo scopo dell'entità
 - Se il corpo di un metodo è troppo grande, scomporre il metodo in più metodi

Documentazione del codice

- Se i metodi o gli attributi di una classe sono troppo numerosi, scomporre la classe in classi con meno responsabilità
- Se la classe svolge uno specifico ruolo di un pattern, specificare il ruolo, il pattern e gli altri partecipanti (quali altre classi e che ruolo hanno); non descrivere il funzionamento del pattern.
- Se il codice rimane comunque di difficile comprensione, inserire commenti in linguaggio naturale.

Aiuto implementativo

- IP per effettuare l'analisi della validità delle parole e per vederne il significato si avvale del dizionario della lingua italiana utilizzato da LibreOffice
 - <https://extensions.libreoffice.org/extensions/italian-dictionary-thesaurus-hyphenation-patterns>
- E' necessario scaricare il file *dict-it.oxt* contenente il dizionario.
- In allegato viene fornito il codice sorgente di alcune classi Java che esemplificano come possa essere effettuata la ricerca di un vocabolo nel suddetto dizionario (la classe principale è Loader) interagendo con il file scaricato.
- Per gestire l'interazione è suggerito di ispirarsi a tale codice, adattandolo alle esigenze di IP.

Aiuto implementativo

- L'invio di email richiede di avvalersi di servizi forniti da un server SMTP.
- JavaMail è una libreria che supporta l'interfacciamento di applicazioni Java con servizi di gestione della posta elettronica, tra cui quelli forniti da server SMTP
- **Usare JavaMail per supportare l'interazione tra serverIP e il server SMTP usato in ateneo**
- In allegato trovate codice sorgente per effettuare la spedizione di una mail usando il server SMTP usato in ateneo.
 - *Usare tale codice come template.*

Aiuto implementativo

NB. Il server SMTP di ateneo consente l'invio di email esclusivamente ad utenti autenticati:

- Le credenziali di autenticazione sono le stesse usate per il login al servizio di web mail dell'università (ad es. username: nome.cognome@studenti.uninsubria.it pwd: mypassword).
- La spedizione è autorizzata solo se le credenziali sono valide e il mittente del messaggio corrisponde allo username.
- **Al fine di gestire in modo sicuro le credenziali, queste dovranno essere richieste in fase di avvio dell'applicazione, evitando la loro memorizzazione su memoria di massa**

Suddivisione del lavoro

- Indicare come è stato suddiviso il prodotto software.
 - Ad es. il programma è stato diviso in “moduli”: GUI, gestione/elaborazione dati, etc.
- Indicare come è stato organizzato il processo di sviluppo (ad es. quali fasi): analisi del problema, modellazione (o design) UML, implementazione, testing, progettazione DB, realizzazione DB, documentazione.
- Indicare per ogni attività l’elenco degli strumenti utilizzati.
- Indicare a chi sono state allocate le diverse attività
 - Ad es. l’analisi è stata fatta da X, i modelli UML da X e Y in coppia, l’implementazione della GUI da X, l’implementazione della gestione dati da Y, etc.

Valutazione

- Il progetto può essere svolto da gruppi formati al massimo da quattro studenti (consigliato gruppi da 3/4).
- **La valutazione è individuale**
- **Lo studente deve saper argomentare in modo opportuno le scelte progettuali, algoritmiche e implementative adottate.**
- **Verrà verificata l'effettiva padronanza delle tecniche utilizzate attraverso una serie di domande.**

Valutazione

- La valutazione terrà conto dei seguenti fattori:
 - l'aderenza del sistema realizzato ai requisiti proposti,
 - i documenti di analisi e progettazione prodotti sia per la realizzazione del software che per il database (correttezza sintattica, semantica, completezza e leggibilità, minimalità dello schema logico),
 - le scelte algoritmiche e di progettazione effettuate (design pattern),
 - la qualità del codice sorgente prodotto (funzionalità, correttezza, facilità d'uso).

Consegna

- Il progetto deve essere inoltrato mediante l'apposito form disponibile sulle pagine del corso sulla piattaforma di elearning
 - Alla voce commenti, specificare nome e matricole degli studenti del gruppo
 - Il progetto deve essere allegato alla mail in formato compresso ZIP.
 - I progetti devono essere corredati da:
 - Documentazione completa
 - Codice sorgente
 - File di build per Ant/Maven per compilare, lanciare il server e i client, creare il db, la documentazione javadoc, etc.
 - Eventuali librerie necessarie alla compilazione e/o all'esecuzione
 - file README con indicazioni sull'installazione e sulla compilazione, specificando i comandi Ant/Maven da utilizzare, ed indicazioni di particolari librerie, usate in modo non standard.
- NB. Il progetto deve compilare correttamente (tramite il file di build), una volta espanso in una directory.
- I progetti con errori di compilazione e/o di esecuzione non verranno valutati.

Discussione

- Gli studenti che hanno effettuato la consegna saranno informati per email sulla data in cui si terrà la discussione orale sul progetto svolto

Domande sul progetto

- Per qualsiasi dubbio durante lo svolgimento del progetto:
pietro.colombo@uninsubria.it
- L'oggetto della mail deve essere: "Laboratorio B: Domande"