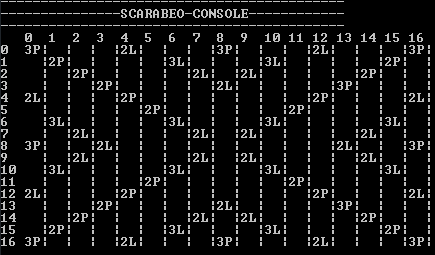
****

**1. Analisi dei requisiti**

**1.1 Scopo del gioco**

**1.2 Punteggio**

**1.3 Tessere**

**1.4 Regole**

**1.5 Svolgimento del gioco**

**2. Progettazione della soluzione**

**2.1 I file del progetto**

**2.2 Classe Casella**

**2.3 Classe Giocatore**

**2.4 Classe Gioco**

**3. Note sulla realizzazione**

**4. Prove di test**

**1. Analisi dei requisiti**

**1.1 Scopo del gioco**

Scarabeo è un gioco da due a quattro giocatori che si svolge in una griglia di 17x17 caselle, nella quale ogni giocatore può immettere una parola da due ad otto caratteri, a patto che essa esista nel vocabolario della lingua italiana e che incroci almeno una lettera già inserita in precedenza. Il gioco termina quando la griglia è piena oppure quando un giocatore finisce le proprie tessere.

Al termine, vince il giocatore che ha totalizzato più punti totali.

**1.2 Punteggio**

Il punteggio viene calcolato in base alle lettere usate dal giocatore. Le lettere hanno i seguenti punteggi:

**1 punto** per ogni A, C, E, I, O, R, S, T;

**2 punti** per L, M, N;

**3 punti** per P;

**4 punti** per B, D, F, G, U, V;

**8 punti** per H e Z;

**10 punti** per Q.

Un caso particolare è la lettera J, che sta ad indicare un Jolly, il quale prende il punteggio della lettera che sostituisce.

Inoltre, esistono 4 tipi di moltiplicatori all’interno della griglia di gioco:

**2L**, che raddoppia il punteggio della lettera presente in quella casella;

**3L**, che triplica il punteggio della lettera presente in quella casella;

**2P**, che raddoppia il punteggio totale della parola inserita;

**3P**, che triplica il punteggio totale della parola inserita.

Infine, vengono assegnati dei punti extra se vengono rispettate alcune condizioni riguardo la lunghezza e la composizione delle parole (questi punti extra NON sono cumulabili, eccetto il primo elencato):

**10 punti extra** se la parola non contiene un Jolly;

**10 punti extra** se la parola è lunga 6 lettere;

**30 punti extra** se la parola è lunga 7 lettere;

**50 punti extra** se la parola è lunga 8 lettere;

**100 punti extra** se la parola è “scarabeo” oppure “scarabei”

**1.3 Tessere**

Scarabeo è formato da un totale di 130 tessere composte dalle varie lettere dell’alfabeto italiano (sono escluse le lettere internazionali K, W, Y e X) che variano in numero a seconda della loro frequenza all’interno delle parole del vocabolario. Le lettere sono così divise:

**12 tessere** per A, E, I, O;

**7 tessere** per C, R, S, T;

**6 tessere** per L, M, N;

**4 tessere** per B, D, F, G, P, U, V;

**2 tessere** per H, Q, Z e J (i due Jolly)

**1.4 Regole**

- Il nome di un giocatore deve essere compreso tra 3 e 32 caratteri. Sono ammessi anche caratteri speciali.

- I giocatori devono necessariamente incrociare una delle lettere inserite in precedenza, ad eccezione del giocatore che ha iniziato la partita. Ci deve essere un’alternanza tra parole inserite orizzontalmente e parole inserite verticalmente.

- Se un giocatore sostituisce un Jolly all’interno della griglia, questo diventa di proprietà del giocatore stesso e può essere ri-usato.

- Sono ammesse sigle di qualsiasi genere riguardo le parole formate da due lettere. Inoltre, non sono consentite lettere accentate: se si vuole inserire una parola che termina o che contiene una lettera con un accento (ad esempio “volontà) è sufficiente comporre la parola usando una lettera non accentata.

- Sono ammessi i verbi declinati, anche con presenza di prefissi e suffissi.

- Le parole da inserire devono appartenere al vocabolario della lingua italiana.

- Non sono ammessi nomi propri di persona.

- Le tessere sono distribuite casualmente: i giocatori ne avranno sempre a disposizione 8 a testa. Dopo aver inserito una parola, un giocatore riceverà automaticamente nella sua mano tante tessere quante ne ha usate per comporla.

- Il gioco termina quando sono state distribuite tutte le tessere, e quando uno dei giocatori ha terminato le proprie.

- L’ultimo giocatore ad aver composto una parola guadagna anche la somma di tutte le altre tessere non ancora usate dagli altri giocatori.

**1.5 Svolgimento del gioco**

1) Si sceglie in quanti giocatori si vuole giocare (da 2 a 4) e quindi si scelgono i nomi da usare.

2) Il giocatore immette una parola nella griglia, facendo sì che questa incroci almeno una lettera presente all’interno della griglia se non è vuota. Nel caso in cui la parola non possa essere immessa in una certa posizione o non incrocia alcuna lettera, sarà possibile re-immettere tutti i dati.

3) Viene calcolato il punteggio della parola, e viene aggiunto a quello del giocatore, per poi distribuire al giocatore stesso delle nuove tessere, in sostituzione a quelle che sono state usate. Se le tessere del gioco sono finite, non ne viene distribuita alcuna.

4) Se il giocatore ha ancora delle tessere nella propria mano, il turno passa al giocatore successivo.

5) I passi da 2 a 4 si ripetono fino a quando uno dei giocatori non è riuscito a immettere tutte le proprie tessere all’interno della griglia di gioco.

6) Viene visualizzato il vincitore con il relativo punteggio, e viene chiesto se si vuole fare un’altra partita. Accettando, si ripeteranno i passi da 1 a 5, mentre rifiutando si chiuderà il gioco.

**2. Progettazione della soluzione**

**2.1 I file del progetto**

Il programma si divide in:

- **Un file di testo**, che contiene le parole presenti all’interno del vocabolario italiano ed i verbi declinati. È possibile eventualmente modificarlo, aggiungendo o togliendo parole secondo le regole del gioco, o anche sostituirlo in toto.

- **Un file main.cpp**, necessario per il lancio del gioco. Oltre al costruttore della classe Gioco, contiene anche una funzione **playerSelecter** che permette di selezionare il numero di giocatori (da 2 a 4) prima dell’inizio vero e proprio del gioco.

- **Tre file di classi** per il gioco, rispettivamente una per le caselle della griglia (Casella.h), una per i giocatori (Giocatore.h e Giocatore.cpp) ed una per il gioco vero e proprio (Gioco.h e Gioco.cpp). Queste vengono discusse più in dettaglio qui sotto.

**2.2 Classe Casella**

La classe Casella (l’unica composta solamente da un header, senza un file .cpp) è stata la prima ad essere realizzata, in quanto responsabile della creazione della griglia di gioco.

Ogni casella ha quattro parametri principali:

- **letter**, ossia il carattere presente all’interno della casella. Questo non comprende i vari modificatori di punteggio come 2L o 3P, inoltre all’inizializzazione sarà sembra vuota.

- **type**, ossia il tipo di casella. Questo a sua volta fa parte di un enum omonimo, all’interno del quale sono presenti i 5 tipi effettivi che la casella può assumere: normal (ossia nessun moltiplicatore), dp(2P), dl(2L), tp(3P), tl(3L).

- **score**, che indica il punteggio della casella. Inizialmente è 0, ma viene aggiornato quando viene inserita una lettera all’interno di una casella, inoltre tiene conto dei due moltiplicatori di lettera 2L e 3L.

- **isJolly**, che indica se la casella contiene la lettera J (ossia un Jolly).

Questi vengono sempre inizializzati con valori di default per far risultare la griglia effettivamente vuota prima dell’inizio del gioco.

Esistono setter e getter per letter e type, un setter per lo score ed uno per isJolly (**jolly**) che setta la casella come Jolly. Per lo score inoltre c’è una funzione **calcolaPunteggioLettera** che calcola in automatico il punteggio della casella.

Infine sono stati overloadati gli operatori di assegnamento e di output: il primo permette così di assegnare un valore al tipo Casella ed è utile per l’inserimento delle parole durante lo svolgimento del gioco, il secondo permette di stampare correttamente ogni casella (e di conseguenza la griglia di gioco).

**2.3 Classe Giocatore**

La classe Giocatore è necessaria a definire alcuni aspetti di ogni giocatore partecipante al gioco. E’ possibile inizializzarla senza o con un nome specifico e possiede tre parametri principali:

- **punteggio**, che tiene conto del punteggio attuale del giocatore. Viene aggiornato attraverso la funzione **updatePunteggio** e possiede sia una funzione setter che una funzione getter.

- **tessere**, un vettore di caratteri nel quale sono elencate le tessere in possesso del giocatore considerato. Queste tessere possono essere rimosse oppure aggiunte durante il gioco, in base alle mosse del giocatore, tramite funzione come **removeTessere** o **addTessera** ma non supereranno mai il limite **MAXTASS** fissato al valore 8. Le tessere possono essere mischiate anche attraverso la funzione **shuffle**, che riordina casualmente le tessere all’interno del vettore.

- **nome**, che contiene appunto il nome del giocatore scelto dal costruttore.

La classe Giocatore ha anche una dimensione definita dalla funzione **size**, in base a quante tessere sono possedute dal giocatore.

Fra le altre funzioni della classe, è possibile stampare le info del giocatore durante il proprio turno e controllare se la parola da inserire è effettivamente valida. **parolaComponibile** serve semplicemente per far sì che il giocatore usi solo le tessere in suo possesso, mentre **parolaInDizionario** cerca se la parola immessa in input dal giocatore è presente nel vocabolario della lingua italiana, descritto dal file .txt allegato al progetto. Infine, **parolaValida** è una funzione di verifica per le parole che contengono un Jolly tra i caratteri al loro interno.

Sia la classe Giocatore, che la classe Casella descritta in precedenza sono quindi essenziali per un buon funzionamento del gioco, che viene descritto nell’ultima classe.

**2.4 Classe Gioco**

La classe Gioco è la più vasta delle tre, ed in essa sono condensati la creazione della griglia di gioco e l’inizializzazione del gioco stesso, il suo svolgimento e la sua fine.

Ogni Gioco si compone di quattro parametri principali:

- **turno**, che conta i turni di gioco, e sul quale si fonda il metodo per accedere ad ogni giocatore durante il **gameLoop**, la funzione principale della classe. Ogni volta che vengono distribuite le nuove lettere al giocatore corrente, il turno passa a quello successivo, incrementando il valore della variabile di 1.

- **players**, ossia il vettore che contiene tutte le informazioni riguardanti i giocatori partecipanti. Grazie a questo vettore, è possibile accedere alle funzioni di ogni classe Giocatore durante il gameLoop.

- **griglia**, che rappresenta l’area di gioco, definita da una matrice quadrata di dimensione **N** (17 di default)

- **tessereTot,** che contiene tutte le tessere presenti all’interno del gioco, che vengono poi distribuite all’inizio del gioco e a mano a mano che si prosegue.

Esiste solamente un costruttore per Gioco, che però comprende le modalità a 2,3 e 4 giocatori. È necessario solamente indicare il numero di giocatori per poter iniziare ed inserire i nomi voluti. Il costruttore inoltre inizializza automaticamente la griglia attraverso **inizializzaGriglia**, con le relative caselle speciali 2P, 3P, 2L e 3L presenti nelle posizioni esatte di partenza, e riempie il vettore tessereTot delle lettere necessarie con la funzione **inizializzaTessere**. In ultimo, lancia la funzione **gameLoop,** che contiene l’intero ciclo dello Scarabeo. Qui vengono distribuite le tessere tramite **distribuisciTessere**, per poi portare il giocatore davanti ad un menu di scelta, rappresentato dalla funzione **sceltaGiocatore**. Il giocatore può scegliere se effettuare una fra sei operazioni disponibili, fra cui l’inserimento di una parola nella griglia, la stampa della griglia di gioco stessa o delle proprie informazioni, l’aiuto per scegliere la miglior parola disponibile fra quelle possibili con i tasselli posseduti, lo shuffle delle tessere oppure la ricerca di una casella jolly. Da notare che, una volta inserita la parola nella griglia, non sarà più possibile tornare indietro.

Avvengono quindi prima i controlli sull’input utente e quindi sulla parola da inserire all’interno della griglia (**paroleIncrociate, parolaComponibile** e **parolaInDizionario**).

In particolare, quando è il momento di inserire una parola che si incrocia con un’altra nella griglia di gioco, le lettere che si incrociano vengono aggiunte temporaneamente alle tessere possedute dal giocatore, per poi venire scartate una volta che la parola è stata inserita tramite **inserisciParolaInGriglia**. Durante il ciclo di vita di questa funzione, viene anche calcolato il punteggio della parola inserita, tramite **calcolaPunteggioLettera** che calcola il punteggio singolo della lettera; queste vengono sommate insieme e infine viene aggiunto al risultato finale l’eventuale funzione **extraPoints**, la quale viene chiamata solamente se la parola è più lunga di cinque lettere.

Infine, all’interno di questa classe è presente anche l’helper (**findBestWord**), il quale interrogato dall’utente riguardo la miglior parola componibile con le lettere in possesso dal giocatore, cerca sul dizionario le parole che è possibile comporre con quelle tessere, le confronta per punteggio e infine ritorna la miglior parola possibile. Questa è indipendente dalle lettere inserite nella griglia di gioco, ed è possibile che non possa essere nemmeno inseribile.

**3. Note sulla realizzazione**

Sono state usate diverse librerie per agevolare la realizzazione del progetto:

- **iostream** per la stampa e l’interazione tra l’utente ed il programma;

- **string e vector** con la prima che è stata usata per la composizione delle parole e per altre variabili; vector è servita principalmente per contenere le informazioni dei giocatori e per poter scorrere con pochi problemi tra i tasselli posseduti dai giocatori stessi;

- **algorithm** per l’algoritmo di random\_shuffle;

- **fstream** per la lettura del file .txt necessario a controllare le parole nel vocabolario;

- **time.h** per generare valori casuali durante la distribuzione delle tessere;

Altre note di progettazione:

- Durante i controlli sulle parole nella griglia, in funzioni come **paroleIncrociate** oppure **inserisciParolaInGriglia**, si è deciso di spezzare i casi in cui la parola veniva inserita verticalmente oppure orizzontalmente.

- Lo shuffle delle lettere è stato inserito per comodità, non essendo una funzione richiesta inizialmente.

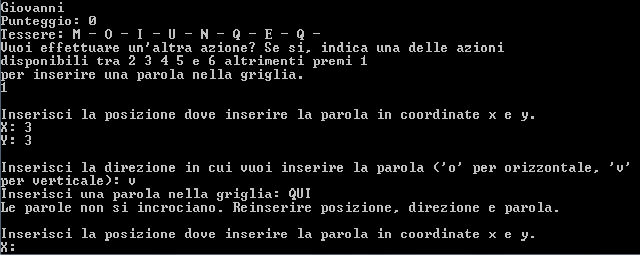
**4. Prove di test**

****

*La schermata principale dopo l’inserimento dei nomi dei giocatori. Vediamo le possibili scelte che può compiere il giocatore, le sue informazioni e la griglia di gioco.*

****

*Inserimento di una parola all’interno della griglia*

****

*Esempio in cui la parola viene inserita in una posizione non valida: il gioco stampa un messaggio in cui è presente l’errore commesso dal giocatore, e permette di re-inserire la parola finché questa non viene inserita nella griglia di gioco.*

****

*La conclusione di una sessione di gioco. Viene stampato il numero del giocatore, il suo nome ed il suo punteggio totale. Successivamente viene chiesto se si vuole giocare di nuovo, tramite una scelta su 0 e 1.*