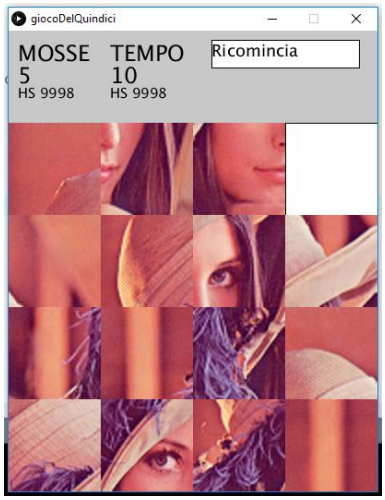
LEGGIMI



La directory contiene uno sketch in Processing che propone una soluzione al progetto assegnato il 10/02/2016 nell'ambito dell'insegnamento di Interazione e Multimedia dell'Università di Catania.

I file del progetto sono così strutturati:

- **giocoDelQuindici.pde** contiene le funzioni base dello sketch

- **gioco.pde** contiene la classe *gioco* che implementa le funzionalità del gioco

- **highscores.txt** contiene i punteggi migliori dell'utente

Il codice è stato scritto con una tendenza verso la programmazione ad oggetti.

La classe gioco implementa i seguenti metodi:

- **int[] generaGrigliaRandom()** genera una griglia di gioco in maniera casuale, e ne verifica la risolubilità, restituendo subito la griglia o eseguendosi ricorsivamente fino a quando non viene trovata una griglia risolubile.

- **void disegnaGriglia()** disegna la griglia (proprietà della classe) sullo sketch. Se è presente un'immagine ne disegna le varie sezioni sulle caselle corrispondenti.

- **void caricaImmagine()** fa comparire una finestra di dialogo per permettere all'utente di scegliere una propria immagine, e passa il risultato ad una funzione *callback* che però non è presente nella classe *gioco* ma è collocata nella classe principale perché altrimenti Processing non riesce a trovarla.

- **void click(int x, int y)** restituisce la casella della griglia (in notazione da 0 a 16) corrispondente alle coordinate passate come parametri.

- **void muovi(int c)** prende come parametro il numero di una casella (in notazione da 0 a 16) e, dopo aver verificato se essa è spostabile, effettua la mossa e chiama un'altra funzione per controllare se la nuova configurazione è vincente.

- **boolean soluzioneVincente()** restituisce *true* o *false* in base al fatto che la grigia (proprietà della classe) sia in una configurazione vincente o meno.

- **void vittoria()** esegue le operazioni di routine di vittoria, come fermare il timer e controllare il miglior punteggio, e se si è battuto, memorizzare le nuove informazioni sul disco. Notare come il salvataggio delle informazioni sia diversificato per mosse e tempo quindi ad esempio se si battono solo le mosse ma non il tempo, verranno registrate le nuove mosse e conservato il vecchio tempo.

- **void iniziaPartita()** esegue le operazioni per l'inizio di una nuova partita, come resettare e far partire il timer, resettare il contatore delle mosse e chiamare la funzione di generazione di una nuova griglia casuale.

- **void scambia(int x, int y)** scambia i contenuti di due caselle della griglia, passate come argomenti, e incrementa il numero di mosse.

- **void scorriTempo()** incrementa la proprietà tempo per contare un nuovo secondo.

Sono inoltre presenti una serie di metodi *get* e *set* che, sebbene non necessari, permettono una più elegante gestione dei dati privati della classe.

Nota sugli highscore: essi vengono mostrati solo quando l'utente ha effettivamente giocato ed ottenuto un numero di mosse e un tempo. Se non si vuole risolvere il rompicapo si può modificare il file *highscores.txt* manualmente. Lo sketch mostrerà gli highscore se essi sono inferiori a 9999.