



Problema 14

15 Maggio 2024

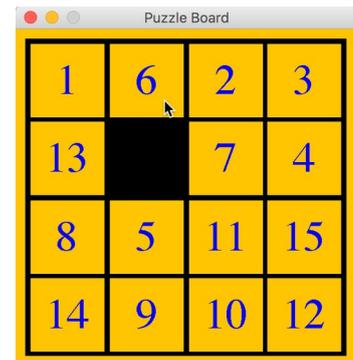
Parte I

Il “rompicapo del 15” è un gioco molto conosciuto. Si tratta di una tavoletta in cui sono collocati 15 tasselli numerati, secondo una matrice 4×4 , in modo che resti un quadratino vuoto (o *lacuna*: vedi figura). A partire da una disposizione disordinata, l’obiettivo è riordinare i tasselli attraverso una sequenza di mosse in ciascuna delle quali uno dei tasselli adiacenti alla lacuna viene fatto scorrere su di essa.

Il compito di programmazione orientata agli oggetti in Java che ti viene assegnato è di modellare la tavoletta del rompicapo, *generalizzandola* al caso $n \times n$.

Innanzitutto, definisci un protocollo appropriato, che comprenda:

- Un costruttore per istanziare un modello della tavoletta;
- Un metodo per verificare se i tasselli sono ordinati;
- Un metodo per verificare se un dato tassello può essere spostato;
- Un metodo per mostrare in forma testuale (stringa) la configurazione;
- Un metodo per spostare un dato tassello.



Ai fini delle operazioni indicate sopra, un tassello viene identificato dal numero corrispondente, riportato su di esso, cioè un numero compreso nell’intervallo $[1, n^2-1]$ per una tavoletta $n \times n$.

Caratterizza la rappresentazione interna della (generica) configurazione di una tavoletta, introducendo le variabili di istanza che ti sembrano adatte per modellare i possibili stati del rompicapo.

Completa quindi la definizione della classe codificando il costruttore e i metodi del protocollo.

Parte II

Una volta realizzato, il modello può essere integrato con uno strumento che consente un’interazione più immediata. Il package *puzzleboard* (jar-file `puzzleboard.jar`¹) mette a disposizione una classe `PuzzleBoard`, per creare una piccola GUI (Graphical User Interface) come quella raffigurata sopra, utilizzabile attraverso il seguente protocollo:

```
PuzzleBoard gui = new PuzzleBoard( n )      crea la rappresentazione grafica di un tavoletta  $n \times n$ 
gui.setNumber( i, j, k )                   colloca in posizione  $\langle i, j \rangle$  un tassello con numero  $k$ 
gui.clear( i, j )                          cancella il tassello in posizione  $\langle i, j \rangle$ 
                                           (in corrispondenza alla quale si crea una lacuna)
gui.display()                              aggiorna la visualizzazione
                                           (i due metodi precedenti non lo fanno automaticamente)
int k = gui.get()                           restituisce il numero associato al tassello su cui si è
                                           agito attraverso un click del mouse
                                           (l’esecuzione di get resta sospesa fino a quando l’utente
                                           attua una selezione valida con il mouse)
```

Gli indici che rappresentano le coordinate dei tasselli sono definiti nell’intervallo $[1, n]$.

Una classe che utilizza questo package deve premettere l’istruzione: `import puzzleboard.PuzzleBoard;` negli ambienti di sviluppo *BlueJ* o *DrJava* è necessario indicare il jar-file nell’ambito dei *classpath* (delle librerie) attraverso le preferenze dell’applicazione; poi si procede come di consueto.

Eventualmente, i comandi per la compilazione e per l’esecuzione da “terminale” sono i seguenti:

```
javac -classpath "puzzleboard.jar:." MainClass.java // MainClass usa PuzzleBoard
java -classpath "puzzleboard.jar:." MainClass
```

(Con il sistema operativo *MS Windows* i due punti vanno sostituiti dal punto e virgola: `"puzzleboard.jar;."`)

L’integrazione delle funzionalità del package *puzzleboard* consente di sperimentare il modello della tavoletta giocando con il rompicapo. In ogni caso, non ti viene richiesto di scrivere un programma che riesca a risolvere il puzzle (compito difficile), ma semplicemente di scrivere un programma che consenta all’utente di giocare agendo con il mouse.

¹ Il package è associato a questo problema nella sezione dedicata al laboratorio delle pagine online del corso.