Informazioni sul documento

Nome documento	Relazione Prima Parte del progetto PuzzleSolver
Versione documento	v.1.0.0
Data redazione	2014-12-04
Redattore	Tesser Paolo

Sommario

Lo scopo del documento è quello di fornire una presentazione del prima parte del progetto PuzzleSolver da realizzare, descrivendo e motivando le scelte attuate in questa fase.

Indice

1 Principi di Programmazione ad Oggetti

In questa capitolo vengono descritte le scelte effettuate per implementare i principi della programmazione ad oggeti, in particolare quelli di incapsulamento e di information hiding.

1.1 Principio di modularità

TO DO

1.2 Principio di incapsulamento

TO DO

1.3 Principio di information hiding

TO DO

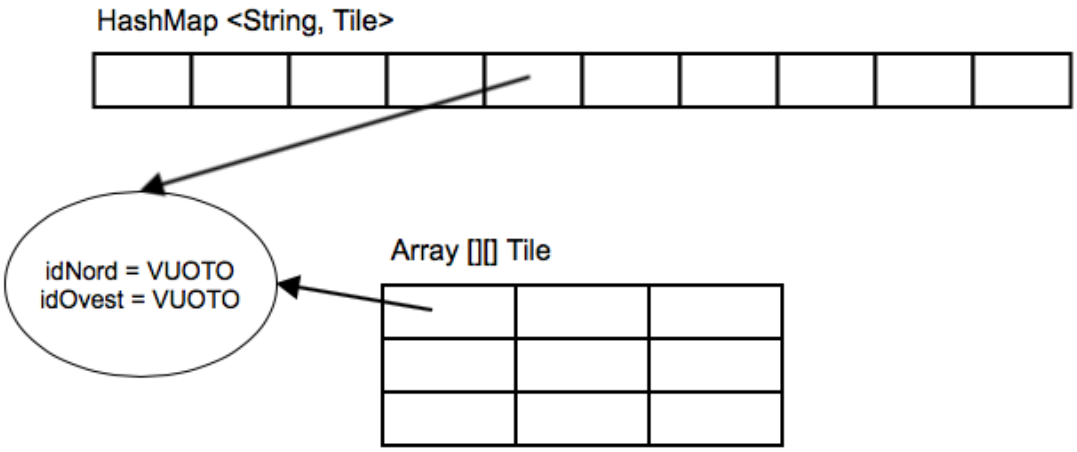
2 Organizzazione delle classi

TO DO

3 Algoritmo di risoluzione

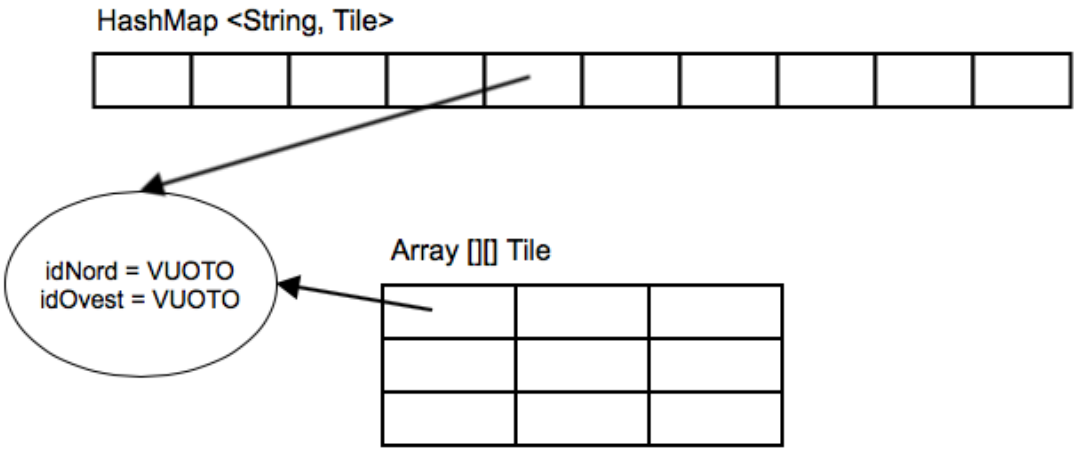
L'algoritmo scelto per risolvere il puzzle è sequenziale, come richiesto dalla specifica di progetto.
Per arrivare alla soluzione vengono utilizzati due strutture dati come membri della classe PuzzleCharacter.
La prima struttura è la collezione HashMap, nella quale salverò in ordine casuale i tasselli ricevuti in input dal file di testo. Memorizzerò dunque l'id del tassello come chiave mentre come valore salverò l'intero pezzo (Tile).
La seconda struttura dati è un array bidimensionale di oggetti Tile. TO DO
Di seguito vengono esposte le seguenze che vengo eseguite, correlate da dei grafici che mostrano come esso agisca sulle strutture dati utilizzate.

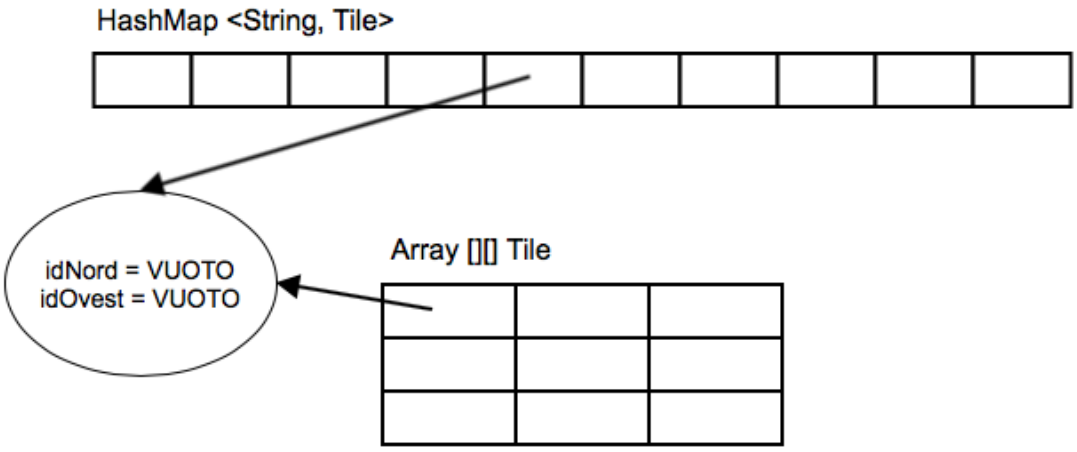
1. **Ricerco il primo elemento del puzzle (quello in alto a sinistra).**
   Per fare ciò scorro una sola volta la tavola hash per cercare il tassello che ha id nord e id ovest uguale alla stringa VUOTO.
   Una volta trovato salvo il pezzo nella prima posizione dell'array bidimensionale.



2. **Ordino la colonna più a sinistra (quella con i tasselli aventi id ovest uguale alla stringa VUOTO).**
   Prendo l'elemento trovato al passo successivo, mi estraggo l'id sud di esso e tramite i metodi della tavola hash mi ricavo il tassello reale a cui corrisponde. Una volta estratto lo inserisco nella posizione corretta.
   Continuo così da quello appena trovato fino a quando non trovo tutti quello sottostanti.

3. **Ordino tutte le righe.**
   Dopo aver ricavato tutta la prima colonna, eseguo secondo lo stesso principio anche la risoluzione per le righe, procedendo però sta volta con la ricerca del pezzo successivo a destra tramite l'id est.


HashMap <String, Tile>
idNord = VUOTO
idOvest = VUOTO
Array [][] Tile


HashMap <String, Tile>
idNord = VUOTO
idOvest = VUOTO
Array [][] Tile

4 Test di correttezza

TO DO