

CORSO DI LINGUAGGI DI PROGRAMMAZIONE

Progetto A.A. 2008/2009

Alberto Ceselli

Maggio 2009

Tantrix Puzzles

Tantrix è un board game ideato in Nuova Zelanda tra la fine degli anni ottanta e la prima metà degli anni novanta. Il gioco è composto dalle 56 pedine rappresentate in Figura 1.

Ogni pedina ha forma esagonale. Nella parte frontale raffigura tre linee di colori diversi (Figura 3(a)). I colori disponibili sono giallo, verde, rosso e blu. Ogni linea collega due facce dell'esagono, ed ogni faccia è l'estremo di una sola linea. Per semplicità, per *colore di una faccia* intendo il colore della linea che ha un estremo su quella faccia. Nella parte posteriore, ogni pedina è identificata da un intero tra 1 a 56 e da un colore (giallo, verde, rosso o blu) (giallo in Figura 3(c)).

Il set di pedine permette, in realtà, di giocare a numerose varianti del Tantrix, che consistono sia in giochi tipo solitario che in sfide da 2 a 4 persone.

Ogni variante richiede di considerare un sottoinsieme del set di pedine e di posizionarle, una di fianco all'altra, anche ruotandole (come in Figura 3(b)), rispettando alcune regole. La regola principale, comune a tutte le varianti del Tantrix, è detta *regola aurea*. E' una generalizzazione della regola di corrispondenza dei simboli delle pedine nel gioco del domino: *ovunque le pedine si tocchino, i colori di tutte le facce a contatto devono combaciare*. La disposizione delle pedine in Figura 4(a), ad esempio, è permessa, perchè tutte le facce che si toccano hanno lo stesso colore; al contrario, la disposizione delle pedine in Figura 4(b) non è permessa, perchè la faccia inferiore sinistra della pedina 3 è gialla, mentre la faccia superiore destra della pedina 1 è verde.

Descrizione del progetto

Il progetto consiste nel realizzare un programma in grado di risolvere automaticamente alcune varianti tipo solitario del Tantrix.

1. **Tantrix domino.** Si sceglie un sottoinsieme delle 56 pedine; il gioco consiste nel posizionare tutte le pedine in sequenza, una di fianco all'altra, in modo che il colore della linea che termina sulla faccia destra di ogni pedina coincida con il colore della linea che termina sulla faccia sinistra della pedina successiva nella sequenza, e simmetricamente che il colore della linea che termina sulla faccia sinistra di ogni pedina coincida con il colore della linea che termina sulla faccia destra della pedina precedente nella sequenza (Figura 4). E' ammesso ruotare arbitrariamente ciascuna pedina.

Figure 1: Il set di pedine di Tantrix

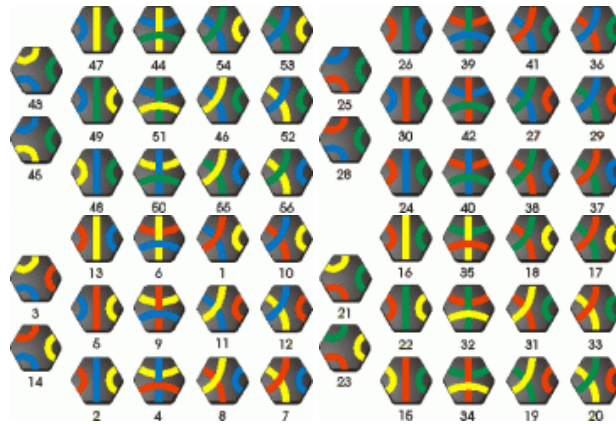


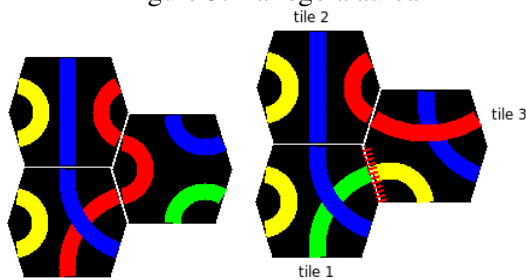
Figure 2: Parte frontale e posteriore di una pedina di Tantrix



(a) Frontale (b) Frontale Ruotata di 60 gradi (c) Posteriore

2. **Tantrix Rainbow.** (a) Considerare il sottoinsieme delle 10 pedine con retro blu. Costruire una piramide (4 livelli) in modo da formare una linea blu, che attraversi senza mai interrompersi tutte le 10 pedine (Figura 5). (b) Considerare il sottoinsieme delle 15 pedine con retro rosso. Costruire una piramide (5 livelli) in modo da formare una linea rossa, che attraversi senza mai interrompersi tutte le 15 pedine. E' ammesso ruotare arbitrariamente ciascuna pedina.
3. **Tantrix board building.** Si sceglie un intero N , e si considera il sottoinsieme di pedine identificate dagli interi da 1 a N . Si sceglie anche una dimensione orizzontale X ed una dimensione verticale Y , e si crea una griglia di X colonne e Y righe. Il gioco consiste nel posizionare tutte le pedine del sottoinsieme selezionato nelle celle della griglia, rispettando la regola aurea (Figura

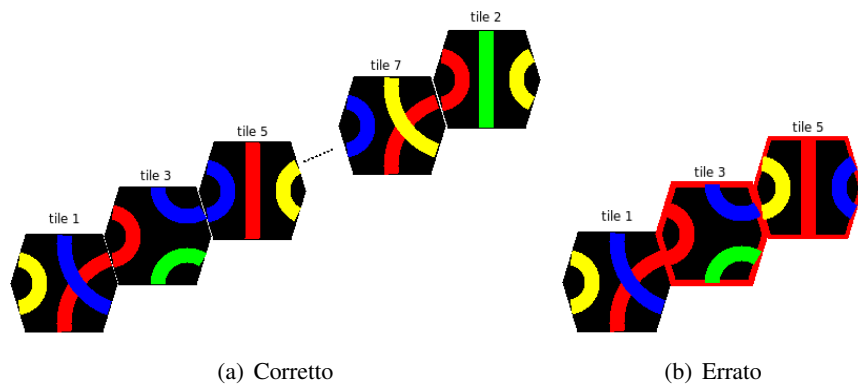
Figure 3: La regola aurea



(a) Corretto

(b) Errato

Figure 4: Tantrix Domino.

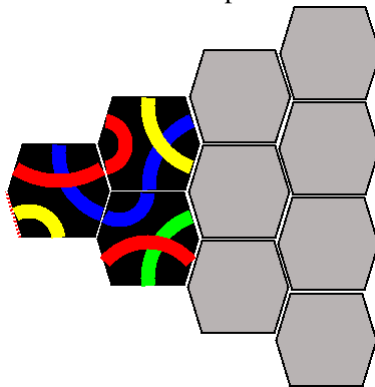


6). E' ammesso ruotare arbitrariamente ciascuna pedina.

4. **Tantrix puzzle.** Come nel board building, si sceglie un intero N , e si considera il sottoinsieme di pedine identificate dagli interi da 1 a N . Si sceglie anche una dimensione orizzontale X ed una dimensione verticale Y , e si crea una griglia di X colonne e Y righe. Il gioco consiste nel posizionare tutte le pedine del sottoinsieme selezionato nelle celle della griglia, (a) rispettando la regola aurea, (b) evitando di creare *hole* sulla griglia e (c) formando una linea che attraversi senza mai interrompersi tutte le pedine del sottoinsieme considerato (es. Figura 8(a)). Per *hole* si intendono sia zone della griglia tra gruppi isolati di pedine (come in figura 6), che posizioni vuote della griglia adiacenti a tre o più pedine (come in Figura 8(b)).

Il puzzle originale prevede le seguenti regole aggiuntive: (d) la linea deve essere *chiusa* e (f) la linea deve essere del colore indicato sul retro della pedina identificata dall'intero N . E' ammesso ruotare arbitrariamente ciascuna pedina.

Figure 5: Tantrix Rainbow: piramide di 10 pedine.



Modalità di svolgimento e criteri di valutazione del progetto

Il programma dovrà essere sviluppato in tre versioni, con tre linguaggi di programmazione diversi:

- prima versione: secondo paradigma di programmazione imperativo, modulare o con approccio orientato agli oggetti (es. utilizzando C, C++, Java, C#, Python ...);
- seconda versione: secondo paradigma di programmazione funzionale (es. utilizzando Scheme, Lisp, ML ...);
- terza versione: secondo paradigma di programmazione dichiarativo o logico (es. Prolog).

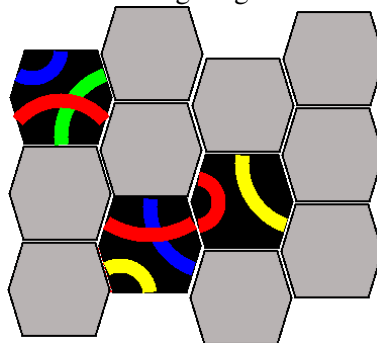
I quattro tipi di solitario richiedono la realizzazione di programmi dalla complessità crescente: *non è assolutamente necessario risolvere tutti i puzzle*. Per ogni tipo di solitario, è possibile affrontare sia un problema di *ricerca* (realizzare un programma che trova *una* soluzione) che un problema di *enumerazione* (realizzare un programma che elenca *tutte le possibili soluzioni*). Un programma che risolve correttamente *anche un solo tipo di puzzle* (anche il più semplice) è considerato sufficiente.

Durante la realizzazione del progetto è necessario privilegiare *lo stile di programmazione e la qualità del codice* realizzato: ai fini del progetto è di secondaria importanza l'efficienza dell'implementazione.

Il programma deve essere corredato di una relazione

- di poche pagine (3 - 4 pagine sono sufficienti)
- che *non* includa codice
- che descriva *marginalmente* gli algoritmi utilizzati
- che descriva *nel dettaglio*
 - le scelte implementative (es. strutture dati utilizzate)
 - le features tipiche del linguaggio o del paradigma che sono state utilizzate in ognuno dei tre programmi
 - eventuali difficoltà incontrate nella realizzazione di una versione rispetto ad un'altra
 - un confronto critico delle tre implementazioni (es. quale è più semplice da realizzare, perchè, quale è più leggibile, quale è più estendibile e modificabile ecc.)

Figure 6: Tantrix board building scegliendo $N = 3$, $X = 4$ e $Y = 3$.



Programmi e relazione devono essere consegnati al docente (tramite email) almeno una settimana prima della prova orale.

La valutazione del progetto terrà conto

- della *qualità* del codice realizzato,
- della pertinenza delle considerazioni sul confronto tra implementazioni, riportate nella relazione,
- di quale tipo di puzzle è stato risolto,
- del tipo di linguaggio scelto per l'implementazione (verranno premiate scelte di linguaggi di programmazione 'Esotici', o per lo meno diversi da C, C# e Java).

Il set di pedine è descritto nel file `tiles.txt`, allegato a questo documento. Nel file sono riportate due tabelle. La prima tabella è composta da due colonne, che indicano rispettivamente l'intero che identifica la pedina e la sequenza di colori delle sulle facce (in senso orario). La seconda tabella è composta da due colonne, che indicano rispettivamente l'intero che identifica la pedina ed il colore riportato nella parte posteriore della pedina.

Nel file `tiles_pl.txt` (sempre allegato a questo documento) è riportata una versione della tabella già formattata per l'utilizzo in un programma Prolog.

Figure 7: Tantrix puzzle: esempio di soluzione corretta e hole (proibito).

