Linguaggi di Programmazione
Corso di Laure in Matematica
Progetto: Partita di Scacchi

docente: Ivano Salvo, Sapienza Università di Roma

7 gennaio 2016

Scopo del progetto è quello di scrivere un programma C++ che gestisce una partita a scacchi, valutando la correttezza delle mosse e controllando se si verificano le condizioni di fine partita.

1 Il gioco degli Scacchi

La Figura 1 ha l'ambizione di riassumere i movimenti di tutti i pezzi degli scacchi[1].

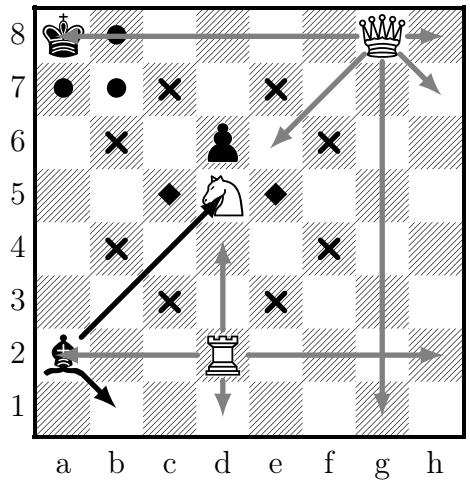


Figura 1: Movimento dei pezzi degli scacchi.

Ogni pezzo può muovere dalla casella in cui si trova in un'altra casella della scacchiera. Se un pezzo P può muovere in una certa casella c, diremo che c è *attaccata* da P. Il Re (nero, casella $a8$) si muove su caselle adiacenti in tutte le direzioni (pallini neri in figura). L'alfiere (nero, casella $a2$) si muove lungo le diagonali (frecce nere). La torre (bianca, casella $d2$) si muove lungo righe e colonne (frecce grigie). Il movimento di torre e alfiere si arresta quando un altro pezzo ne ostacola il movimento. Se il pezzo è un pezzo avversario, questo può essere *catturato* (o *preso*) ed eliminato dal gioco[2]. Nel nostro esempio, la torre bianca può catturare l'alfiere nero (occupandone la posizione), e l'alfiere nero può catturare il cavallo bianco. Il cavallo (bianco, casella $d5$) si muove con un caratteristico andamento ad "L" percorrendo una casella in una direzione (orizzontale o verticale), e due nell'altra (caselle contrassegnate con una 'x'). Cattura come gli altri pezzi, ma a differenza di torre e alfiere, il suo movimento non viene intralciato da eventuali altri pezzi sulla sua traiettoria (ad esempio il pedone nero in figura). Il pedone, che può andare solo avanti (di un passo, a meno che non si trovi nella sua casella di partenza, quando può scegliere di avanzare di 2 passi), si muove lungo la colonna, ma attacca le due caselle adiacenti in diagonale (rombi neri in figura) e può catturare pezzi avversari che si trovino lì. Infine, la donna (bianca, casella $g8$) ha il movimento combinato di torre e alfiere. Diremo, inoltre che il re è *sotto scacco* se si trova in una casella attaccata da un pezzo nemico. Nel nostro caso, il re nero è sotto scacco a causa della donna bianca.

A volte si usa il termine *semi-mossa* per indicare la mossa di un colore, e *mossa* per indicare una coppia di mosse (una del bianco e una del nero). Noi useremo sempre il termine mossa per indicare una semi-mossa.

1.1 Fine della Partita

La partita può finire per i seguenti motivi:

Scacco Matto Si verifica quando il Re del giocatore a cui tocca muovere è sotto scacco e non c'è nessuna mossa che permette al Re di sottrarsi dallo scacco. Il colore che subisce lo scacco matto perde la partita (è di fatto, l'obiettivo del gioco);

Patta La partita è patta nelle seguenti situazioni:

- **Stallo:** Il Re del giocatore a cui tocca muovere *non* è sotto scacco e non c'è nessuna mossa legale possibile (o perchè i pezzi sono bloccati, o perchè eventuali mosse metterebbero il Re sotto scacco).
- **Ripetizione di Posizione** Si ripete la stessa posizione, con mossa allo stesso giocatore per tre

[1] Per ulteriori informazioni sulle regole del gioco potete consultare, ad esempio, l'esauriente voce di Wikipedia, `http://it.wikipedia.org/wiki/Scacchi`

[2] Per i non scacchisti: il Re non può mai essere catturato!

volte (questa situazione viene spesso impropriamente chiamata *ripetizione di mosse*, ma le posizioni potrebbero ripetersi anche a distanza di molte mosse). A volte si chiama *scacco perpetuo* il caso particolare in cui uno dei due Re, pur non subendo mai scacco matto, non può sottrarsi a continui scacchi di pezzi avversari.

Materiale Insufficiente: si verifica quando sulla scacchiera non sono rimasti abbastanza pezzi affinchè uno dei due giocatori possa dare scacco matto. Ad esempio Re e Alfiere contro Re. Semplificando un po', potete far terminare il programma per patta quando entrambi i giocatori hanno al più una figura (Cavallo o Alfiere) oltre al Re. È interessante notare invece, che Re e Pedone possono vincere contro il Re in quanto il Pedone può promuovere a Donna o Torre.

50 mosse di entrambi i giocatori senze prese e senza mosse di pedone.

Abbandono Uno dei due giocatori decide di abbreviare la propria agonia rendendo onorevolmente le armi all'avversario nel momento in cui giudica che non ci sono più speranze. È il modo più tipico in cui finiscono le partite con esito fatale.

2 Specifiche del progetto

Voi dovete implementare una classe `PartitaDiScacchi` che ha lo scopo di memorizzare una posizione del gioco degli scacchi e verificare la legalità delle mosse. Inoltre, tale classe dovrà saper leggere una sequenza di mosse, stampare la posizione corrente, e la sequenza di mosse giocate. Possibilmente, dovrà anche verificare se si è verificata una situazione di fine partita (patta, scacco matto, abbandono).

Più precisamente, essa dovrà implementare almeno i seguenti metodi:

`PartitaScacchi()` costruttore che inizializza la partita con la posizione iniziale del gioco degli scacchi e settando il turno al bianco;

`PartitaScacchi(String[] p)` costruttore che inizializza la partita con la posizione specificata dall'array di stringhe `p`; sotto viene spiegato come specificare una posizione;

`muovi(Casella da, Casella a)` verifica se è possibile muovere il pezzo presente nella posizione specificata dal parametro `da` nella casella specificata dal parametro `a`, in accordo col turno corrente. Il metodo `muovi` deve, all'occorrenza, comunicare in modo opportuno, aualora la mossa non sia legale, il motivo: sollevare le seguenti eccezioni checked, nel caso in cui la mossa effettuata non sia legale:

Casella Partenza Vuota se la casella `da` è vuota;

Casella Arrivo Occupata se la casella `a` è già occupata da un pezzo del colore che sta muovendo;

Turno Errato se la casella `da` contiene un pezzo di colore diverso rispetto al giocatore di turno;

Re Sotto Scacco se la mossa non è possibile in quanto il Re del colore che ha mosso risulta essere sotto scacco dopo l'esecuzione della mossa;

Scacco Matto si deve bloccare la partita se si riscontra che la partita finisce a causa di uno scacco matto (ovviamente può avere dato scacco matto solo il giocatore che ha mosso);

Mossa Illegale in ogni altro caso di mossa illegale (esempio: la casella `a` non è raggiungibile dal pezzo che si trova nella casella `da`);

`abbandona()` Il giocatore di turno rinuncia a continuare e la partita finisce con la vittoria dell'altro giocatore;

`ritira()` che permette di tornare indietro di una mossa. Benchè sia proibito ai tornei ritirare le mosse (anzi, anche i pezzi toccati vanno obbligatoriamente mossi, purchè possano fare mosse legali) e non sia sportivo ritirare le mosse neanche in contese amichevoli, questo metodo è utile ad esempio per poter rivedere la partita o anche come "metodo di servizio" (ad esempio nella verifica dello scacco matto, si potrebbe trovare semplice eseguire tutte le possibili mosse e verificare se il Re non puo' sottrarsi allo scacco. Dopo l'esecuzione queste mosse vanno ritirate). Osserviamo, che nel caso di prese è necessario conoscere anche il pezzo avversario che è stato catturato, e quindi dovete tenerne conto nella memorizzazione delle mosse.

`void print()` stampa la posizione corrente, con indicazione di chi tocca muovere (vedi Sez. 4).

Potrebbero infine essere utili (ai fini della verifica dello scacco matto) metodi per calcolare la lista delle mosse possibili, per un certo pezzo, per tutti i pezzi di un certo colore, e per tutti i pezzi.

3 Struttura del Codice

Il codice dovrà il più possibile essere strutturato ad oggetti, allo scopo di rendere il più facile possibile il suo adattamento a variazioni del gioco degli scacchi (ad esempio scacchi eterodossi con pezzi dal movimento diverso rispetto a quelli tradizionali) e al suo riuso per implementare altri giochi su scacchiera. In particolare si invitano gli studenti a:

- rappresentare la scacchiera come una matrice di oggetti di tipo `Pezzo` e rappresentare le caselle vuote col pointer `NULL`;
- Il tipo `Pezzo` è una classe astratta che prevede un metodo `boolean puoiMuovere(Casella a, Posizione p)` che determina se il pezzo che risponde al messaggio può eseguire la mossa (ciò può dipendere dalla posizione, motivo del secondo parametro). Così facendo è semplice considerare varianti del gioco degli scacchi in cui ci sono pezzi che hanno regole di movimento diverse o anche altri giochi (ad esempio la dama o simili);

4 Codifiche

Mosse: La classe `Casella` semplicemente può essere una classe che contiene due interi, uno che rappresenta la riga e uno la colonna.

Diversa la codifica da usarsi nell'input-output per le mosse. Potete semplicemente usare due interi anche nell'input output.

Per chi avesse desiderio di fare un progetto più carino, la codifica più ragionevole è la così detta "notazione algebrica": ogni casella viene identificata da una lettera (da a ad h per le usuali scacchiere 8×8) che identifica la colonna e un numero (da 1 a 8) che identifica la riga. Ogni mossa si può rappresentare senza abiguità con una stringa che identifica la casella di partenza e quella d'arrivo del pezzo che si muove[3]. Ad esempio, nella scacchiera di Fig. 1 il cavallo in $d5$ può andare in $c7$: tale mossa si indica con la stringa `d5-c7`. Usualmente per le prese, si usa il carattere `x` in luogo del trattino, per cui, sempre in figura, la mossa con cui la torre prende l'alfiere verrebbe indicata con la stringa `d2xa2`.

Posizioni: Una scacchiera (per voi è necessario solo negli output) può essere rappresentata da 8 stringhe di 8 caratteri. Una stringa può contenere i caratteri: `T, t, C, c, A, a, D, d, R, r, P, p, .`, dove:

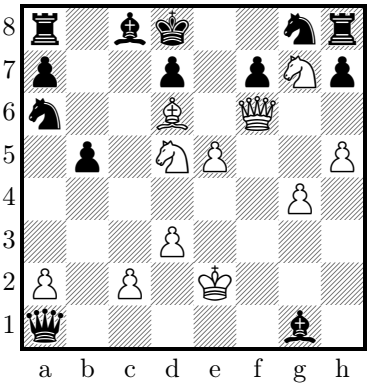
- `.` rappresenta una casella vuota;
- le lettere maiuscole rappresentano caselle occupate pezzi bianchi e le lettere minuscole rappresentano pezzi bianchi;
- `T` e `t` rappresentano (casella occupata da) una torre, `C` e `c` rappresentano un cavallo, `A` e `a` rappresentano un alfiere, `D` e `d` rappresentano una donna, `R` e `r` rappresentano un re.

[3]per chi avesse letto almeno un libro di scacchi, questa **non** è la notazione usuale, in cui si preferisce accettare qualche ambiguità in favore della leggibilità, indicando solo la casella d'arrivo e la natura del pezzo che si muove. Le ambiguità sono dovute al fatto che ad esempio due torri o due cavalli (o due donne e due alfieri in posizioni un po' grottesche) potrebbero portarsi nella stessa casella.

La seguente figura mostra un esempio di scacchiera codificata e relativa posizione.

Codifica:

```
t.ar..ct
p..p.pCp
c..A.D..
.p.CP..P
......P.
...P....
P.P.R...
d.....A.
N
```



Notazione FEN: Eventuali appassionati, possono studiarsi la notazione FEN che rappresenta in forma compatta una posizione del gioco degli scacchi. Ad esempio la posizione sopra in figura viene rappresentata dalla stringa `r1bk2nr/p2p1pNp/n2B1Q2/1p1NP2P/6P1/3P4/P1P1K3/q5b1`. Qualcuno capisce perchè[4]?

5 Test ed Interfaccia Testuale

Il funzionamento della classe `PartitaScacchi` può essere testato semplicemente da una funzione (un `main` per esempio) che genera una nuova istanza e poi invia una sequenza di messaggi `muovi`, `ritira`, etc.

Per ottenere un effetto più gradevole, è possibile definire una interfaccia testuale (che accetta ad esempio mosse in notazione algebrica e genera i corrispondenti messaggi `muovi(Casella da, Casella a)` e stampa ad ogni mossa la situazione della partita come descritto sopra.

Chi avesse spiccati interessi di programmazione, può anche cimentarsi in una interfaccia grafica.

In ogni caso si tratta di strati di software separati che *usano* la classe `PartitaScacchi`.

6 Regole particolari

Negli scacchi esistono alcune "mosse particolari" (arrocco, promozione, presa en-passant), che fanno eccezione all'usuale struttura delle mosse e che di conseguenza potrebbero dare piccoli problemi di implementazione. La loro implementazione è quindi da ritenersi facoltativa e lasciata agli studenti interessati al gioco o alle sfide di programmazione.

[4]un aiutino: trattandosi di una notazione internazionale, i pezzi hanno le iniziali inglesi: K per King, Q per Queen, B per Bishop (=Vescovo, cioè Alfiere), N per kNight (Cavaliere, cioè Cavallo), R per Rook (Torre) e P per Pawn (Pedone).