

Gocce di Java

Progetto Stratego Duel

Pierluigi Crescenzi

Università degli Studi di Firenze

20 Aprile 2016

- ▶ In alto a sinistra: riga 0 e colonna 0
- ▶ In basso a destra: riga 9 e colonna 9

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4			●	●			●	●		
5			●	●			●	●		
6										
7										
8										
9										

- ▶ Caselle verdi non sono utilizzabili

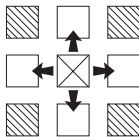
- ▶ Due giocatori
 - ▶ Rosso: primo a giocare
 - ▶ Blu: secondo a giocare
- ▶ 10 carte (ciascuna con un suo punteggio) per ogni giocatore
 - ▶ 1 bandiera con nome FL e senza punteggio
 - ▶ 2 bombe con nome FB e SB e senza punteggio
 - ▶ 1 maresciallo con nome MA e punteggio 10
 - ▶ 1 generale con nome GE e punteggio 9
 - ▶ 2 artificieri con nome FM e SM e punteggio 3
 - ▶ 2 esploratori con nome FS e SS e punteggio 2
 - ▶ 1 spia con nome SP e punteggio 1

- ▶ Rosso posiziona le carte nelle prime quattro righe in alto
 - ▶ Ovvero nelle caselle dalla (0,0) alla (3,9)
- ▶ Blu posiziona le carte nelle ultime quattro righe in basso
 - ▶ Ovvero nelle caselle dalla (6,0) alla (9,9)
- ▶ Esempio tipico (ma prevedibile)

	0	1	2	3	4	5	6	7	8	9
0								SM	SB	FL
1								SS	SP	FB
2								MA	GE	FM
3										FS
4			●	●			●	●		
5			●	●			●	●		
6	FS									
7	FM	GE	MA							
8	FB	SP	SS							
9	FL	SB	SM							

- ▶ Nel gioco reale, le carte sono coperte
 - ▶ Nel gioco “software” ciò è gestito dal GameManager
- ▶ Le due righe centrali sono inizialmente vuote

- ▶ I giocatori giocano a turno: il rosso inizia
- ▶ A ogni suo turno, un giocatore deve muovere una carta in avanti, indietro, a sinistra o a destra di una posizione



- ▶ Le carte Flag e Bomb non possono muoversi
- ▶ Lo Scout può muoversi di più di una posizione (nella stessa direzione), purché tutte le caselle intermedie siano libere e non siano una delle otto caselle verdi
- ▶ La casella di arrivo non può essere occupata da una carta del giocatore che fa la mossa
- ▶ La casella di arrivo non può essere una delle otto caselle verdi
- ▶ Se la casella di arrivo è occupata da una carta del giocatore avversario, ha inizio uno *scontro*

- ▶ Vince la carta con il punteggio più alto
 - ▶ La carta che perde viene rimossa dalla tavola da gioco
 - ▶ Se il punteggio è uguale, entrambe le carte vengono rimosse
- ▶ Regole speciali
 - ▶ Se la carta che si è mossa non è `Miner` e la carta “attaccata” è `Bomb`, la carta che si è mossa perde e viene rimossa dalla tavola da gioco
 - ▶ Se la carta che si è mossa è `Miner` e la carta “attaccata” è `Bomb`, la carta che si è mossa vince, `Bomb` viene rimossa dalla tavola da gioco e la carta che si è mossa occupa la sua posizione
 - ▶ Se la carta che si è mossa è `Spy` e la carta “attaccata” è `Marshall`, la carta che si è mossa vince, `Marshall` viene rimossa dalla tavola da gioco e la carta che si è mossa occupa la sua posizione

- ▶ Non è consentito continuare a spostare una pedina ininterrottamente avanti e indietro fra due caselle
 - ▶ Nella versione software, una carta può tornare nella stessa casella solo dopo che la carta ha fatto almeno 3 mosse

- ▶ Vince il primo giocatore che raggiunge la bandiera dell'avversario
- ▶ Vince anche un giocatore il cui avversario non ha più mosse da fare
- ▶ Nella versione software, la partita sarà terminata (in parità) dopo 200 mosse per giocatore

- ▶ `public void start(boolean isFirst);`
 - ▶ Comunica al giocatore se gioca per primo (ovvero se è il giocatore rosso)

- ▶ `public int[] position(String piece);`
 - ▶ Valori possibili di piece: "FL", "FB", "SB", "MA", "GE", "FM", "SM", "FS", "SS", "SP"
 - ▶ Restituisce riga e colonna in cui posizionare la carta
- ▶ Esempio con posizionamento visto in precedenza
 - ▶ `position` invocato al giocatore rosso con argomento "SB"
 - ▶ Valore ritornato: array {0,8}
 - ▶ `position` invocato al giocatore blu con argomento "MA"
 - ▶ Valore ritornato: array {7,2}

- ▶ `public void viewPositions(int[] [] position);`
 - ▶ Comunica 4 righe iniziali dell'avversario (0 libera/1 occupata)
 - ▶ Giocatore rosso: da casella (0,0) a casella (3,9)
 - ▶ Giocatore blu: da casella (6,0) a casella (9,9)
- ▶ Esempio con posizionamento visto in precedenza
 - ▶ `viewPositions` invocato al giocatore rosso con argomento l'array
 - `{0,0,0,0,0,0,0,0,1,1,1}, {0,0,0,0,0,0,0,0,1,1,1},`
`{0,0,0,0,0,0,0,0,1,1,1}, {0,0,0,0,0,0,0,0,0,0,1}}`
 - ▶ `viewPositions` invocato al giocatore blu con argomento l'array
 - `{{1,0,0,0,0,0,0,0,0,0,0}, {1,1,1,0,0,0,0,0,0,0,0},`
`{1,1,1,0,0,0,0,0,0,0,0}, {1,1,1,0,0,0,0,0,0,0,0}}`

- ▶ `public int[] move();`
 - ▶ Restituisce riga, colonna, direzione e numero di passi
 - ▶ Direzione: 1 (avanti), 2 (indietro), 3 (sinistra), 4 (destra)
 - ▶ Numero di passi maggiore di 1 solo se la carta da muovere è Scout
- ▶ Esempio con posizionamento visto in precedenza
 - ▶ Una possibile risposta del giocatore rosso potrebbe essere l'array

$$\{2,8,1,1\}$$

- ▶ Corrisponde a muovere General di una posizione in avanti: la nuova posizione di General sarà (3,8)
- ▶ Una possibile risposta del giocatore blu potrebbe essere l'array
$$\{6,0,1,5\}$$
 - ▶ Corrisponde a muovere primo Scout di cinque posizioni in avanti: la nuova posizione di primo Scout sarà (1,0)

- ▶ `public void tellMove(int[] move);`
 - ▶ Comunica riga, colonna, direzione e numero di passi della mossa dell'avversario
- ▶ Esempio con mosse viste in precedenza
 - ▶ `tellMove` invocato al giocatore blu con argomento l'array
`{2,8,1,1}`
 - ▶ `tellMove` invocato al giocatore rosso con argomento l'array
`{6,0,1,5}`

- ▶ `public void fight(String piece);`
 - ▶ Invocato a entrambi i giocatori solo se la casella raggiunta è occupata, comunica la carta con cui avviene lo scontro
- ▶ Esempio dopo mosse viste in precedenza
 - ▶ Supponiamo che mossa successiva di rosso sia $\{1,7,4,7\}$
 - ▶ `fight` invocato al giocatore rosso con argomento "FS"
 - ▶ `fight` invocato al giocatore blu con argomento "SS"
 - ▶ Ha luogo uno scontro: le carte hanno punteggio uguale e vengono rimosse dalla tavola di gioco

► Il GameManager

- Invoca il costruttore senza parametri dei due giocatori
- Invoca il metodo `start` dei due giocatori con il giusto parametro
- Invoca 10 volte il metodo `position` per ciascun giocatore
- Comunica tramite il metodo `viewPositions` le caselle occupate dall'avversario nelle sue 4 righe iniziali
- Finché la partita non finisce, a turno
 - Invoca il metodo `move` e controlla la correttezza della mossa
 - Comunica la mossa all'avversario mediante il metodo `tellMove`
 - Comunica (mediante il metodo `fight`) l'eventuale carta con cui ha luogo lo scontro e determina il risultato dello scontro
- Ogni irregolarità di un giocatore genera un'eccezione che fa terminare la partita e assegna la vittoria finale all'avversario

- ▶ Classe <Cognome>Player che implementa interfaccia Player
- ▶ Inclusa in pacchetto `gj.stratego.player.<cognome>`
 - ▶ Tutte le classi che usa devono essere in questo pacchetto
- ▶ Il giocatore da console
 - ▶ Classe ConsolePlayer inclusa in pacchetto `gj.stratego.player.console`
 - ▶ Consente di giocare digitando le mosse da tastiera
 - ▶ Utile per debugging

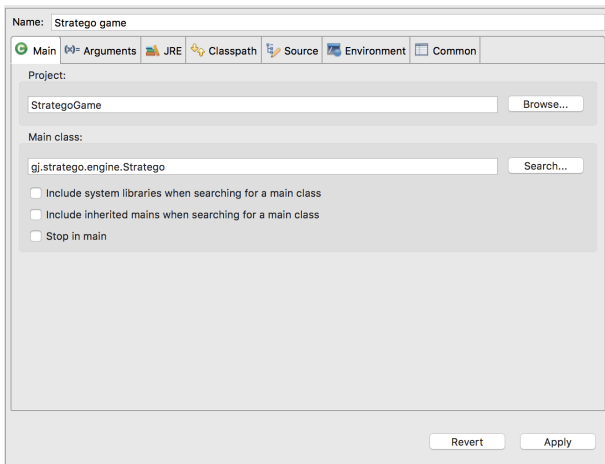
- ▶ Creare un progetto chiamato `StrategoGame`
- ▶ Selezionare la sotto-voce `Folder` della voce `New` del menu `File` e creare una cartella chiamata `lib`
- ▶ Copiare il file `stratego.jar` (disponibile sul sito web del corso) all'interno della cartella `lib`
- ▶ Selezionare la sotto-voce `Add to Build Path` della voce `Build Path` del menu contestuale che si apre cliccando con il tasto destro sul file `stratego.jar`
- ▶ Creare nella cartella `src` il pacchetto `gj.stratego.player.console` e copiare all'interno del pacchetto la classe `ConsolePlayer.java` (disponibile sul sito web del corso)

► Situazione

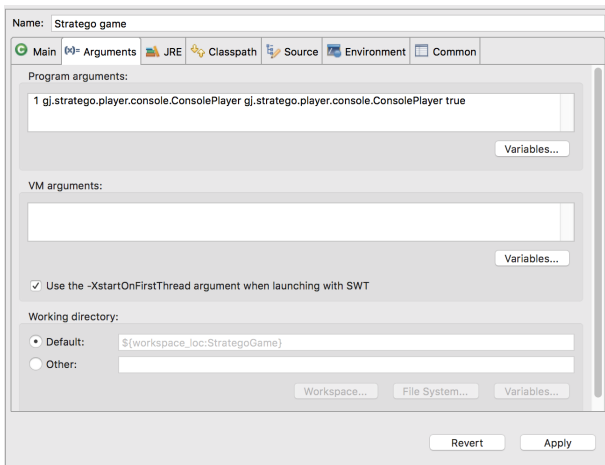


- Selezionare la voce Run configurations... del menu Run e creare una nuova configurazione Java Application: inserire i campi come mostrato nella figura

- Selezionare la voce Run configurations... del menu Run e creare una nuova configurazione Java Application: inserire i valori nei campi come mostrato nella figura



- Selezionare la linguetta Arguments, inserire i valori nei campi come mostrato nella figura e premere Apply



- Premere Run
- La partita ha inizio, viene mostrata la tavola da gioco iniziale e al giocatore rosso viene chiesto di inserire la prima mossa

ConsolePlayer vs ConsolePlayer

Beginning of 1 games

	0	1	2	3	4	5	6	7	8	9
0	FLR	SBR	SMR							
1	FBR	SPR	SSR							
2	FMR	GER	MAR							
3	FSR									
4			L	L			L	L		
5			L	L			L	L		
6	FSB									
7	FMB	GEB	MAB							
8	FBB	SPB	SSB							
9	FLB	SBB	SMB							

Insert red move row: