

Corso di Laurea in Ingegneria Informatica

Lezione 10

Esercizio sulla progettazione

P. Foggia – N. Capuano

DIEM - Università di Salerno





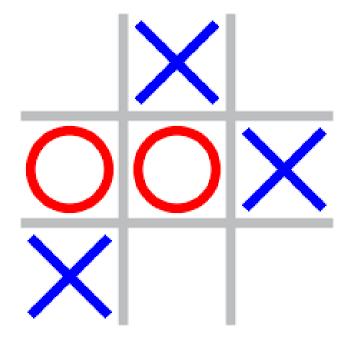
Schema

- Decomposizione in moduli
- Definizione delle interfacce
- Valutazione di coesione e accoppiamento



Il problema

Vogliamo realizzare un programma che consente di giocare al gioco Tic-Tac-Toe (detto anche "Tris")



Il problema

Vogliamo realizzare un programma che consente di giocare al gioco Tic-Tac-Toe (detto anche "Tris")

- Abbiamo come vincolo di progettazione che il programma deve essere realizzato in C
- Il programma deve consentire a un essere umano di giocare contro il computer
 - Non è richiesta una modalità a due giocatori umani
- Non è richiesta un'interfaccia utente di tipo grafico

Il primo passo nella progettazione di dettaglio è la decomposizione in moduli.

Il primo passo nella progettazione di dettaglio è la decomposizione in moduli.

- Decidiamo di seguire un approccio funzionale, quindi partiamo dalle operazioni che il programma deve svolgere
 - Gestione della scacchiera (es. inizializzazione, aggiornamento dopo una mossa, controlli sulla validità di una mossa e sullo stato della partita)
 - Visualizzazione di informazioni all'utente
 - Acquisizione di input dall'utente
 - Calcolo della mossa del computer
 - Un main che richiama le altre funzioni nell'ordine giusto

Cominciamo con una definizione di massima dei moduli.

tictactoe

Il main che implementa la logica della partita

interfaccia_utente

Visualizzazione informazioni e acquisizione input scacchiera

Gestione della scacchiera

strategia

Algoritmo per il calcolo della mossa del computer

• È l'unica decomposizione possibile?

- È l'unica decomposizione possibile?
 - No: ad esempio:

tictactoe

Il main che implementa la logica della partita

visualizzazione

Visualizzazione informazioni all'utente

scacchiera

Gestione della scacchiera

strategia

Algoritmo per il calcolo della mossa del computer

input

Acquisizione input dall'utente

- È l'unica decomposizione possibile?
 - No: ad esempio:

tictactoe

Il main che implementa la logica della partita

strategia

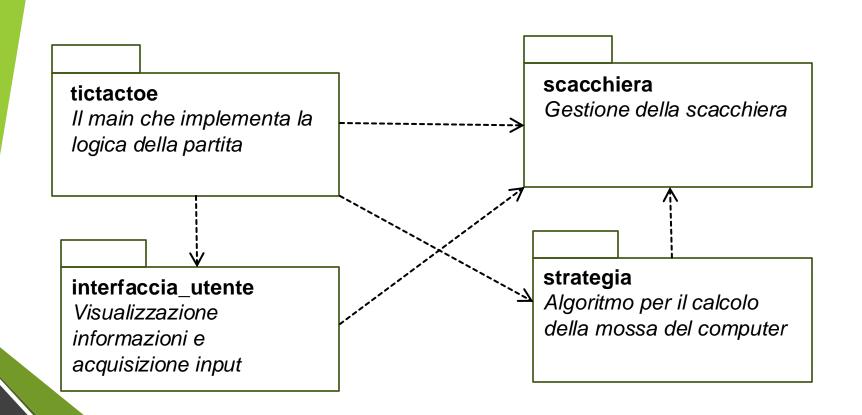
Algoritmo per il calcolo della mossa del computer

scacchiera

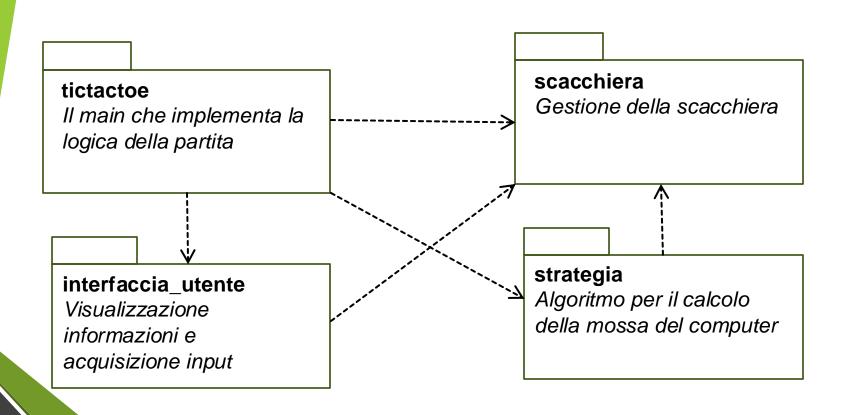
Gestione della scacchiera, inclusa la visualizzazione e l'input della mossa

> discuteremo tra poco l'impatto in termini di coesione e accoppiamento

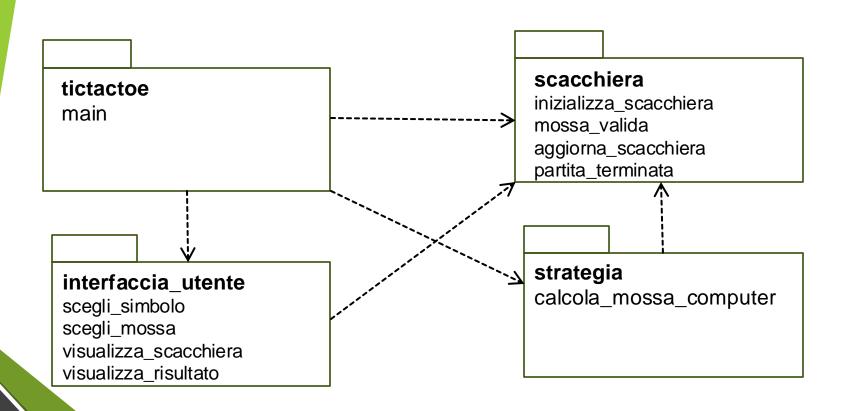
Dipendenze tra i moduli



Cominciamo a individuare le operazioni di ciascun modulo



Cominciamo a individuare le operazioni di ciascun modulo



- Ora dobbiamo definire per ciascuna operazione il prototipo e le altre informazioni del contratto
- Ci serve una struttura dati per rappresentare una scacchiera
- Ci serve un tipo per rappresentare il contenuto di una cella della scacchiera (il simbolo X, il simbolo O oppure la cella vuota)

... dove inseriamo queste definizioni?

Cominciamo a individuare le operazioni di ciascun modulo scacchiera Cella Scacchiera tictactoe inizializza_scacchiera mossa valida main aggiorna_scacchiera partita_terminata strategia interfaccia_utente calcola_mossa_computer scegli_simbolo scegli_mossa visualizza scacchiera visualizza_risultato

Possiamo descrivere le interfacce usando Doxygen

- Per ogni modulo definiamo un file .h (tranne per tictactoe, che non esporta funzioni usate dagli altri moduli, e quindi avrà solo il .c)
- Nel .h inseriamo i prototipi delle funzioni e i commenti di documentazione per Doxygen

Vedi i file html allegati alla lezione per i dettagli delle interfacce...

Valutazione di coesione e accoppiamento

Coesione e accoppiamento

Proviamo a valutare coesione e accoppiamento dei moduli progettati scacchiera Cella Scacchiera tictactoe inizializza_scacchiera mossa valida main aggiorna_scacchiera partita_terminata strategia interfaccia_utente calcola_mossa_computer scegli_simbolo scegli_mossa visualizza scacchiera

visualizza_risultato

Cominciamo con la coesione

tictactoe main

Coesione funzionale Il modulo realizza una singola operazione

strategia calcola_mossa_computer

Coesione funzionale Il modulo realizza una singola operazione

Cominciamo con la coesione

scacchiera

Cella Scacchiera inizializza_scacchiera mossa_valida aggiorna_scacchiera partita_terminata

Coesione funzionale

Il modulo realizza un unico compito: la gestione della struttura dati che rappresenta la scacchiera

Cominciamo con la coesione

interfaccia_utente

scegli_simbolo scegli_mossa visualizza_scacchiera visualizza_risultato

Coesione procedurale

Le funzioni hanno una elevata probabilità di essere usate insieme; però non servono a un unico compito e (tranne scegli_mossa e visualizza_scacchiera) non lavorano sugli stessi dati.

E se avessimo scelto questa alternativa?

tictactoe

Il main che implementa la logica della partita

strategia

Algoritmo per il calcolo della mossa del computer

scacchiera

Gestione della scacchiera, inclusa la visualizzazione e l'input della mossa

E se avessimo scelto questa alternativa?

scacchiera

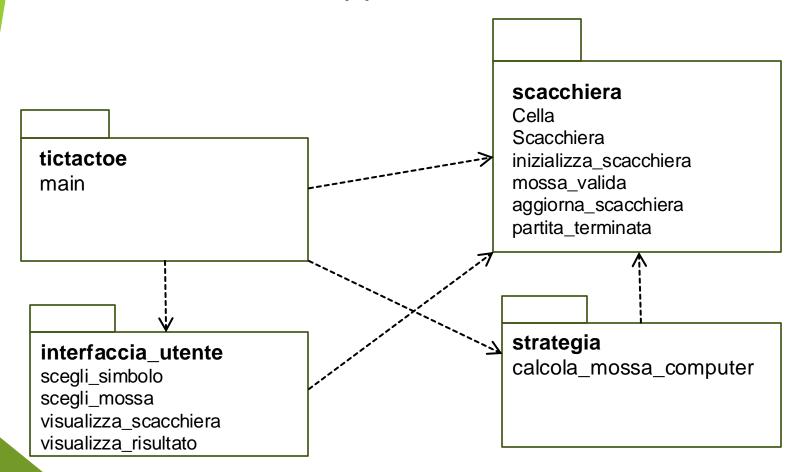
Cella
Scacchiera
inizializza_scacchiera
mossa_valida
aggiorna_scacchiera
partita_terminata
scegli_simbolo
scegli_mossa
visualizza_scacchiera
visualizza_risultato

Coesione procedurale

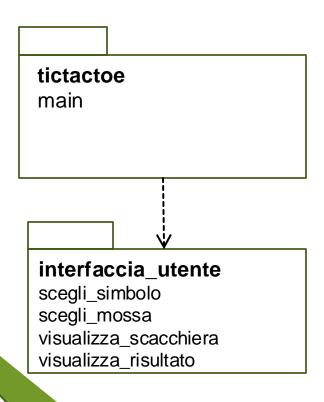
Le funzioni hanno una elevata probabilità di essere usate insieme.

Dal punto di vista della coesione, abbiamo peggiorato rispetto alla prima ipotesi.

Ora valutiamo l'accoppiamento



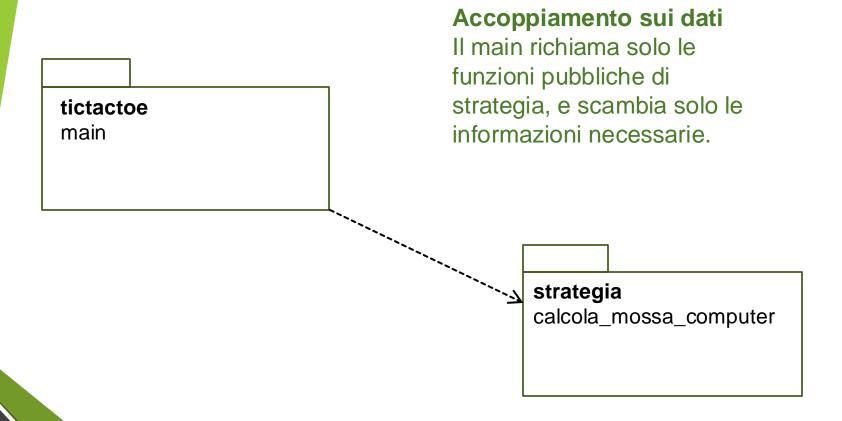
Ora valutiamo l'accoppiamento



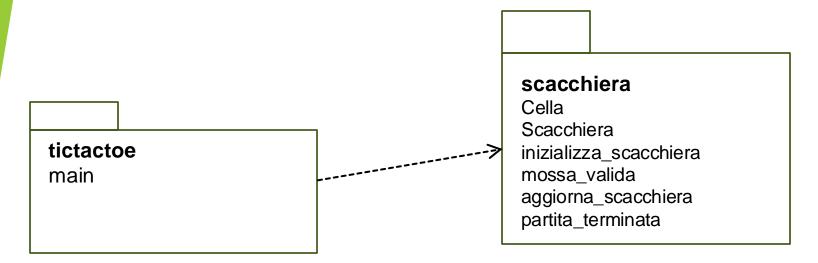
Accoppiamento sui dati

Il main richiama solo le funzioni pubbliche di interfaccia_utente, e scambia solo le informazioni necessarie.

Ora valutiamo l'accoppiamento



Ora valutiamo l'accoppiamento



Accoppiamento sui contenuti

Il main ha bisogno di accedere alla struttura dati di Scacchiera per sapere qual è il prossimo giocatore che deve muovere.

Ora valutiamo l'accoppiamento

Accoppiamento sui contenuti

visualizza_scacchiera ha bisogno di accedere alla struttura dati di Scacchiera per sapere qual è il contenuto di una cella e qual è il prossimo giocatore che deve muovere.

scacchiera

Cella Scacchiera inizializza_scacchiera mossa_valida aggiorna_scacchiera partita_terminata

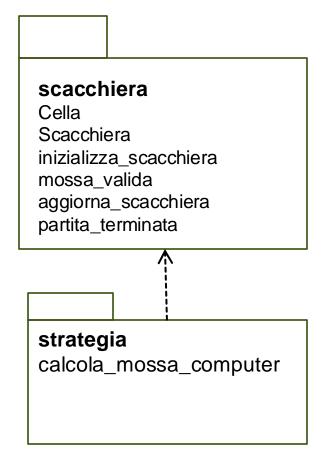
interfaccia_utente

scegli_simbolo scegli_mossa visualizza_scacchiera visualizza_risultato

Ora valutiamo l'accoppiamento

Accoppiamento sui dati

Strategia richiama solo le funzioni pubbliche di scacchiera, e scambia solo le informazioni necessarie.



Possiamo migliorare i problemi di accoppiamento individuati aggiungendo due operazioni al modulo scacchiera:

- Cella prossimo_giocatore(Scacchiera *s);
- Cella contenuto_cella(Scacchiera *s, int posizione);

... lasciamo come esercizio la definizione del loro contratto!

E se avessimo scelto questa alternativa?

tictactoe

Il main che implementa la logica della partita

visualizzazione

Visualizzazione informazioni all'utente

scacchiera

Gestione della scacchiera

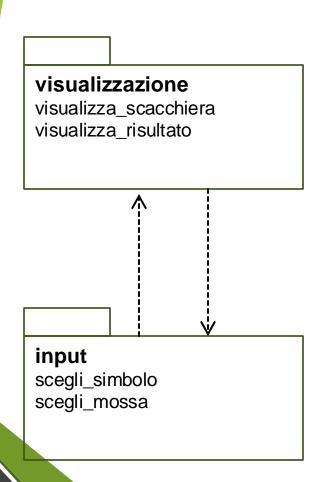
strategia

Algoritmo per il calcolo della mossa del computer

input

Acquisizione input dall'utente

• E se avessimo scelto questa alternativa?



Accoppiamento sui contenuti

Anche se nessuno dei due moduli richiama le funzioni dell'altro, l'implementazione di visualizza_scacchiera e quella di scegli_mossa sono collegate; la modifica di una ha effetto sull'altra e viceversa.