

Introduzione alla Programmazione a.a. 2024/2025

Progetto

October 7, 2024

Questo documento contiene le linee guida per la realizzazione del progetto.

1 Descrizione del Progetto

Si richiede agli studenti di sviluppare un programma C che implementi la logica del seguente gioco di carte:

1.1 Componenti

- Un mazzo di 40 carte diviso in 4 semi
 - 10 carte di Fiori (da 1 a 7, con J,Q K)
 - 10 carte di Picche (da 1 a 7, con J,Q K)
 - 10 carte di Quadri (da 1 a 7, con J,Q K)
 - 10 carte di Cuori (da 1 a 7, con J,Q K)

1.2 Giocatori

- Da 2 a 20 giocatori

1.3 Scopo del Gioco

- Rimanere l'ultimo giocatore in gioco

1.4 Svolgimento del Gioco

All'inizio del gioco ogni giocatore ha **2 PUNTI VITA**.

Il gioco si svolge a fasi. Ad ogni fase:

1. Mescolare il mazzo
2. Distribuire ad ogni giocatore 2 carte: una coperta e una scoperta.
3. Scegliere casualmente il primo giocatore della fase.

1.4.1 Svolgimento della Fase

La fase si svolge dando a ciascun giocatore un turno. A partire dal primo giocatore, in ordine crescente ad anello, il giocatore nel suo turno:

1. Deve risolvere l'effetto della propria carta scoperta (vedi effetto carte).
2. Può visionare la propria carta coperta e opzionalmente scoprirla per applicarne l'effetto. Nel caso fosse già scoperta per effetto di altre carte, non sarà possibile applicarne nuovamente il suo effetto.

3. Il giocatore termina la propria giocata cedendo il turno al giocatore successivo. Alla fine del turno di tutti i giocatori, la fase si conclude e chi non ha più vite viene eliminato.

Cosa significa in ordine crescente ad anello?

- Supponiamo di avere 3 giocatori (G1, G2, G3)
 - Se il primo giocatore è G1, la fase seguirà i turni di $G1 \rightarrow G2 \rightarrow G3$
 - Se il primo giocatore è G2, la fase seguirà i turni di $G2 \rightarrow G3 \rightarrow G1$
 - Se il primo giocatore è G3, la fase seguirà i turni di $G3 \rightarrow G1 \rightarrow G2$

1.5 Effetto delle carte

- **Carte dal 2 al 6:** non succede nulla.
- **Carta 7:** il giocatore in possesso della carta forza il giocatore successivo a scoprire la sua carta coperta e ad applicarne l'effetto.
- **Carta J:** il giocatore in possesso della carta deve dare 1 PUNTO VITA al giocatore precedente.
- **Carta Q:** il giocatore in possesso della carta deve dare 1 PUNTO VITA al secondo giocatore successivo (cioè, saltando un giocatore).
- **Carta 1:** il giocatore in possesso della carta perde 1 PUNTO VITA, che verrà lasciata sul campo di gioco (le vite sul campo di gioco non vengono resettate a ogni fase, quindi rimangono anche per le successive se non sono state rimosse).
- **Carta K:** il giocatore in possesso della carta, riceve tutti i PUNTI VITA lasciati sul campo di gioco fino a quel momento.

1.6 Fine della Partita

Il gioco termina quando rimane solo un giocatore al termine di una fase e gli altri sono stati tutti eliminati.

1.7 FAQ - Regole del Gioco

- **In due giocatori, come funziona l'effetto della Carta Q?** The player who holds that card will lose 1 LIFE POINT and will subsequently receive his own LIFE POINT as there is a ring order of the players.
- **Ad ogni turno, vengono distribuite 2 carte ai giocatori, ma queste sono cumulabili nelle fasi/turni successivi?** No. Tutte le carte alla fine di ogni fase vengono ritirate e mischiate nel mazzo, quindi ridistribuite ad ogni giocatore. I giocatori avranno sempre e solo 2 carte.
- **Se il giocatore corrente non girasse la carta, questa andrebbe scartata?** No, rimane in gioco coperta perchè potrebbe essere attivata da altri giocatori utilizzando regole speciali (es: l'ultimo giocatore del turno se usasse l'effetto della Carta 7, andrebbe ad affliggere il primo giocatore di quel turno, portando potenzialmente all'attivazione a catena di altri effetti di carte coperte).

2 Specifiche di Implementazione

Il programma dovrà chiedere ai giocatori quale mossa giocare, e dovrà visualizzare il campo di gioco aggiornato (usando la funzione `printf` o equivalente). Dovrà verificare che la mossa scelta sia legale (non vada in conflitto con le regole del gioco), e dovrà aggiornare le componenti nelle varie fasi di gioco. Questi passi si ripetono fino alla fine della partita, che verrà notificata su terminale.

Nel caso in cui il giocatore corrente richieda opzionalmente di scoprire una carta che porterebbe alla perdita del giocatore corrente, è richiesto che venga visualizzato un messaggio di avvertenza e

che esso venga confermato (es: "L'azione porta ad azzerare le vite del giocatore corrente. Sei sicuro di voler procedere? [Si/No]").

Inoltre, è possibile presentare delle migliorie al progetto, che potranno essere valutate positivamente. Possibili migliorie potrebbero essere: grafica migliore, menu di interazione dell'utente più usabile, ecc. Non ci sono limiti alle aggiunte che vorrete fare! Quindi non ponetevi limiti!

3 Consegna

Quando? Il progetto deve essere consegnato 4 giorni prima la data di ciascun appello. E' sempre obbligatorio iscriversi all'appello di "Esercizi" del prof. Olivieri per sostenere la discussione del progetto.

Dove? Il docente abiliterà la consegna tramite Moodle secondo le scadenze previste.

Cosa? Dovrete consegnare un unico file zip contenente:

1. Una relazione scritta di al massimo 3 pagine che descriva la struttura del vostro progetto, l'organizzazione del lavoro tra i componenti del gruppo, le principali difficoltà incontrate. Relazioni più lunghe verranno penalizzate.
2. Il codice sorgente del progetto in linguaggio C99; eventuali parti di codice scritto in altri dialetti vanno isolati in sorgenti separati ed il progetto dovrà compilare opportunamente.
3. Documentazione delle funzioni, dei tipi e dei file generata con Doxygen Librerie esterne.
4. È possibile implementare l'intero gioco utilizzando semplicemente la standard library di C, usando le funzioni di lettura dei tasti come `scanf`, `getc` o `getchar` e le funzioni di stampa come `printf`. Coloro che vogliono cimentarsi nell'uso di librerie esterne per dare una veste grafica ed una interazione col giocatore più accattivanti e fluide possono farlo, tuttavia sarà propria responsabilità occuparsi degli aspetti di portabilità e compatibilità. Il programma deve compilare e funzionare correttamente sulle 3 piattaforme (Windows, Linux, Mac); in caso contrario, è necessario documentare e motivare opportunamente la scelta nella relazione che accompagnerà il progetto ed in sede d'esame.

4 Valutazione

Il progetto è da svolgere in gruppi di **max 4 studenti**. La composizione dei gruppi è libera. In casi eccezionali (es. studenti lavoratori, ...), il progetto potrà essere sviluppato anche individualmente se concordato con il prof. Olivieri.

Il progetto sarà valutato in trentesimi e andrà a pesare il 30% del voto finale. Il progetto **dovrà essere sufficiente** (un voto maggiore o uguale a 18) per poter verbalizzare il voto finale.

Il progetto verrà valutato indicativamente come segue:

Progetto di Gruppo

- progetto **sufficiente**, se permette ad almeno 2 utenti di giocare inserendo le proprie mosse tramite input da tastiera (modalità interattiva) e visualizzare tramite output il campo da gioco e la propria mano nel turno corrente; l'implementazione delle regole deve completa e corretta.
- progetto **buono**, se implementa il precedente punto ed implementa una logica iterativa che permetta di aggiungere in modo parametrico il numero di utenti (no 20 utenti hard-coded!) e il numero di default delle punti vita dei giocatori (es: maggiore di 2); Vanno gestiti anche casi speciali come l'inserimento di mosse o valori non validi;
- progetto **ottimo**, se implementa correttamente i precedenti punti e viene fornita una documentazione Doxygen esaustiva del progetto;

Tutti i membri del gruppo devono conoscere ogni riga del codice!

Importante: Il progetto è di gruppo, ma la valutazione è individuale. Questo significa che i componenti di un gruppo potrebbero ricevere un voto diverso.

Progetto Individuale (ad esempio per studenti lavoratori)

- progetto **sufficiente** se permette ad almeno 2 utenti di giocare inserendo le proprie mosse tramite input da tastiera (modalità interattiva) e visualizzare tramite output il campo da gioco e la propria mano nel turno corrente; l'implementazione delle regole del gioco può essere ragionevolmente parziale.
- progetto **buono**, se implementa il precedente punto ed implementa una logica iterativa che permetta di aggiungere in modo parametrico il numero di default delle punti vita dei giocatori (es: maggiore di 2); Vanno gestiti anche casi speciali come l'inserimento di mosse o valori non validi;
- progetto **ottimo**, se implementa correttamente i precedenti punti viene generata e viene fornita una documentazione Doxygen esaustiva del progetto;