Gioco dell’Oca Pazza

**INTRODUZIONE**

Versione digitale del gioco dell’oca con alcune variazioni.

Per la realizzazione del progetto ci siamo serviti della piattaforma GitHub

-GitHub: <https://github.com/jambode/GOP>

**INTERFACCIA**

L’interfaccia utilizzata è quella testuale, il gioco si svolge interamente sul terminale.

**TEMA**

**….**

**CARATTERISTICHE**

* 2 o più giocatori
* 6 tipi di carte:
  + Tira di nuovo;
  + Avanza;
  + Torna indietro;
  + Torna all’inizio;
  + Salta il turno;
  + Rispondi alla domanda;
* Azioni delle caselle:
  + Vai avanti;
  + Vai indietro;
  + Tira ancora il dado;
  + Salta il turno;
  + Pesca una carta;
  + Fortuna;

**COME COMPILARE IL PROGRAMMA**

**(**makefile ??)

**COME ESEGUIRE IL PROGRAMMA**

Per eseguire il gioco basta aprire il file da terminale.

**IMPLEMENTAZIONE**

* **CLASSI**
  1. TABELLONE:
     + Il tabellone ha come parametri un puntatore che gestisce gli spostamenti sulle caselle (implementate come una lista) e
  2. GIOCATORE:
     + I giocatori sono gestiti da due classi:
     + La prima:
       - Nella classe “player” sono presenti i parametri *nome* (array), *turno* (intero);
       - I giocatori sono indicati nel gioco attraverso una sigla, a ciascuno la sua;
       - Ogni giocatore inizia dalla casella “partenza” ;
       - Durante il gioco viene stampata la posizione di ogni partecipante e la rispettiva sigla nella casella in cui si trova;
     + Le seconda:
       - Nella classe “inizializza\_player” troviamo un puntatore ad un oggetto di tipo “player”, un intero per indicare la posizione del singolo giocatore, uno per indicare il numero totale dei partecipanti e un ultimo per il conteggio dei turni. E’ presente anche un array che serve per inizializzare i giocatori con una sigla in modo da poter tenere traccia dei loro spostamenti nel tabellone.
       - La classe si trova quindi a gestire il numero, i nomi dei giocatori ed i turni di questi con i relativi spostamenti
  3. DADO:
     + Il dado è composto da un intero che restituisce i valori da 1 a 6 in modo randomico ogni qual volta si richiami la funzione “lancia\_dado”;
  4. MAZZO CARTE:
     + Il mazzo di carte è un insieme di 40 carte
     + La classe mazzo presenta i parametri “*mazzo\_carte”* (array di classi) e “*num\_cartacorrente”* (intero che indica il numero della carta e va da 1 a 40);
     + Il mazzo viene mischiato ogni qual volte che viene richiamata l’apposita funzione
  5. CARTE:
     + Nella classe “carta” sono presenti parametri *valore* (intero) e *msg* (messaggio di tipo stringa);
     + Ogni carta è indicata con un numero e un messagio/azione;
     + La carta che viene estratta, quando richiesto, è quella che si trova in cima al mazzo e al termine del turno verrà inserita in forno al mazzo;
     + La funzione delle carte è quella di far svolgere delle azioni al giocatore, azioni quali;
       - Vai avanti: il giocatore verrà fatto avanzare di 3 caselle;
       - Vai indietro: il giocatore verrà fatto retrocedere di 3 caselle;
       - Salta turno: il giocatore in questione non potrà giocare fino alla fine del turno;
       - Riparti: il giocatore tornerà alla casella di partenza;
       - Rispondi ad una domanda: al giocatore verrà posta una domanda a risposta multipla, se la risponderà correttamente come premio avanzerà nel tabellone (con le stesse modalità dell’azione “vai avanti” descritta precedentemente), il caso di risposta errata sarà costretto ad indietreggiare secondo un numero di caselle sancito dal lancio del dado;
       - Rilancia: il giocatore ha diritto a lanciare nuovamente il dado e a spostarsi di conseguenza in base al numero indicato da questo;
  6. EFFETTI:
     + La classe degli effetti prende come input un puntatore alla classe dei gicatori, un mazzo di carte e le carte;
     + Nella classe sono contenute sia le azioni delle caselle che quelle delle carte (non so se metterlo)
* **STRUTTURE**
  1. CASELLE DEL TABELLONE:
     + Le caselle del tabellone sono state realizzate come una struttura dinamica tramite l’utilizzo di una lista bidirezionale;
     + La struttura prevede come parametri un intero che indica il numero delle caselle e due puntatori, uno alla casella precedente e uno a quella successiva;
     + Il numero di caselle del tabellone varia ad ogni partita secondo un numero che va da 40 a 60;
     + Le azioni implementate per le caselle sono:
       - Tira di nuovo: il giocatore ha diritto a lanciare nuovamente il dado e a spostarsi di conseguenza;
       - Vai avanti: il giocatore verrà fatto avanzare di un numero di caselle pari a quello risultante dal lancio del dado;
       - Vai indietro: stessa funzionalità dell’azione sovrastante, unica differenza il giocatore verrà fatto indietreggiare anziché avanzare;
       - Pesca carta: verrà estratta la prima carta del mazzo ed eseguita l’azione corrispondente;
       - Salta turno: il giocatore che incappa in questa casella non potrà giocare per il turno successivo;
       - Fortuna: il giocatore verrà spostato su una casella che verrà estratta in modo random tra quelle presenti nel tabellone, purtroppo o per fortuna;
  2. DOMANDE:
     + Il file contiene 20 domande a risposta multipla (4 opzioni) ed una sola risposta corretta;
     + La lista delle domande ha come parametri: la stringa domande, la stringa risposta e il carattere della risposta esatta;
     + La modalità di scelta delle domande è puramente casuale e gestita da una funzione che stampa una di queste in modo random;
     + La struttura è gestita da una funzione booleana che ha lo scopo di mostrare la domanda e le sue opzioni e verificare che la risposta data dal giocatore sia corretta o meno. Nel primo caso il giocatore verrà fatto avanzare di un numero di caselle sancito dal lancio del dado, nel caso della risposta errata verrà fatto indietreggiare con la stessa modalità;

**GESTIONE DEL MAIN (inizio/ fine e scopo del gioco)**

**….**

REALIZZATO DA:

grazia, graziella e grazie al \*\* ☺