

Gioco dell'oca pazza.

POWERED BY  N | Solid

build passing

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

Il tema è l'antico Egitto con i suoi misteri.

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 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:
 - Classe ***“player”***:
 - Qui sono presenti i parametri nome (array di caratteri) e posizione (intero);

- Ogni giocatore ha il proprio nome;
- Ogni giocatore inizia dalla casella “partenza” ;
- Durante il gioco viene stampata la posizione di ogni partecipante e il rispettivo nome nella casella in cui si trova;
- Classe ***"inizializza_player"***:
 - Qui troviamo un puntatore ad un oggetto di tipo “player”, un intero 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 il proprio nome in modo da poter tenere traccia dei loro spostamenti nel tabellone.
 - Nella classe vi sono metodi utili per inizializzare i giocatori inserendone il numero e i rispettivi nomi. Ci sono metodi utili alla gestione dei turni con i relativi spostamenti.
 - Ci sono metodi utili alla creazione e gestione del tabellone (creato attraverso una lista di caselle (struct)).

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 il metodo ***"lancia_dado"***;

4. *CARTE*:

- Nella classe “carta” sono presenti parametri valore (intero) e msg (messaggio di tipo stringa);

- Ogni carta è indicata con un numero e un messaggio/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 fondo 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;
 - Scambia posto con un altro giocatore: il giocatore in questione dovrà scambiare posto con un altro giocatore scelto in modo randomico, può quindi sia tornare indietro che andare avanti.
 - Torna alla casella di partenza: il giocatore tornerà alla casella di partenza;
 - Rispondi ad una domanda: al giocatore verrà posta una domanda a risposta multipla, se 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.;
 - Rilancia: il giocatore ha diritto a lanciare nuovamente il dado e a spostarsi di conseguenza in base al numero indicato da questo;

5. *MAZZO CARTE*:

- Il mazzo di carte è un insieme di 40 carte, inizializzato dal metodo *“crea_mazzo”*.
- La classe mazzo presenta i parametri “carte” (array di *“carte”*) e un puntatore a *“player”*
- Il mazzo viene mischiato ogni qual volta che viene richiamata l'apposita funzione

6. *EFFETTI*:

- La classe degli effetti prende come input un puntatore alla classe dei giocatori, e un oggetto dado.
- Nella classe sono contenute le azioni delle caselle e alcune di quelle delle carte

• 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/indietro: qui il giocatore in modo

random andrà avanti/indietro di un numero casuale;

- 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 avanzare per il turno in questione;
- Vai alla casella n: il giocatore verrà spostato su una casella che verrà estratta in modo random tra quelle presenti nel tabellone, purtroppo o per fortuna;

2. *DOMANDE:*

- La struttura 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:

- Claudia Agostini
- Martina Dell'Elce
- Federico Augelli