

Jan 23, 14 18:46

main.m

Page 1/2

```

%
% Data una matrice M HxW che rappresenta lo stato di una partita di forza 4,
% scrivere una funzione che effettui la ricerca di sequenze di 4 elementi
% consecutivi orizzontalmente o verticalmente. La matrice M e' letta da un file
% ascii 'partita-salvata.txt'
%
% * [stop colore r c dir] = ricerca(M)
%
% dove "stop" e' un logical pari a 1 (vero) solo se una sequenza di colore
% "colore" e' stata trovata in posizione "r" "c" e in direzione "dir" (che
% varra' 'v' se verticale, 'o' se orizzontale, '' altrimenti). In caso
% contrario "stop" e' pari a 0 (continua a giocare).
%
% La funzione dovra' utilizzare le seguenti due funzioni, anch'esse da
% implementare:
%
% * [r c] = ricerca_seq_orizz(M, colore)
% * [r c] = ricerca_seq_vert(M, colore)
%
% dove [r c] conterra' la posizione (riga, colonna) in M della prima occorrenza
% del carattere passato come parametro "colore". Se la sequenza non e' trovata,
% la funzione ritornera' [-1 -1].
%
% Creare poi la funzione seguente
%
% * [M ok] = gioca(M, colore, colonna)
%
% che, se il tabellone lo permette, inserisce una pedina di colore "colore"
% nella colonna specificata. Se la giocata e' possibile allora ritorna "ok" = 1
% (vero) e la matrice M modificata opportunamente. Altrimenti, "ok" sara' zero.
%
% Per implementare "gioca" e' consigliato implementare una funzione
%
% * top = cerca_top(M, colonna)
%
% che, data una colonna, cerca la cima di tale colonna, ossia la prima riga con
% una posizione libera, pari a 'o'. Se non vi e' alcuna posizione libera,
% allora top sara' zero.
%
% Infine, utilizzare le funzioni create per giocare interattivamente al gioco
% del forza 4, informando l'utente sul colore che sta giocando e chiedendo
% all'utente di inserire una colonna valida.
%
% In ogni momento, l'utente deve poter interrompere la partita e salvare lo
% stato della partita in un file 'partita-salvata.mat', che dovra' contenere,
% la matrice M, i nomi dei due giocatori in ordine (corrispondenti ai colori
% 'r' e 'b'), e il colore del prossimo turno ('r' o 'b'). Suggerimento, per i
% nomi usare una struttura e non un vettore di stringhe.
%
% carichiamo lo stato della partita e le variabili
partita_salvata = load('partita-salvata.mat');
M = partita_salvata.M; % accedo al campo "M" della struttura "partita_salvata"
giocatori = partita_salvata.giocatori;
turno = partita_salvata.turno;

% visualizziamo la matrice con la funzione creata nelle precedenti esercitazioni
%
visualizza(M, 1);

% dato il prossimo turno, prepariamo la sequenza delle prossime giocate
if turno == 'r'
    seq = ['r', 'b'];
else % il prossimo turno era il blu
    seq = ['b', 'r'];
end

stop = 0;

% finche' non vale la condizione di stop (ossia, qualcuno ha vinto)
%
while ~stop
    % per ogni colore
    for colore = seq
        if colore == 'r'

```

```

        giocatore = giocatori(1).nome;
    else
        giocatore = giocatori(2).nome;
    end

    salva = input('Vuoi salvare la partita e uscire?', 's');

    if salva == 'Y'
        turno = colore;
        save('partita-salvata.txt', 'M', 'turno', 'giocatori');
        exit;
    else
        ok = 0;

        while ~ok
            % gioca
            colonna = input(sprintf('Turno di "%s" ("%s"). Colonna? ',
                                    giocatore, colore));
            [M ok] = gioca(M, colore, colonna);

            if ~ok
                fprintf('Giocata non valida!\n');
            end
        end

        % la giocata e' stata accettata, visualizziamo la matrice in figura 1
        visualizza(M, 1);
        [stop colore r c dir] = ricerca(M);
    end
end

% TODO: salva una matrice vuota, gli stessi giocatori e il turno del vincitore
fprintf('Ha vinto il colore %c: %c(%d, %d)\n', colore, dir, r, c);

```