



Corso di Informatica Medica

Esercitazione V

28 maggio 2014

Alessandro A. Nacci

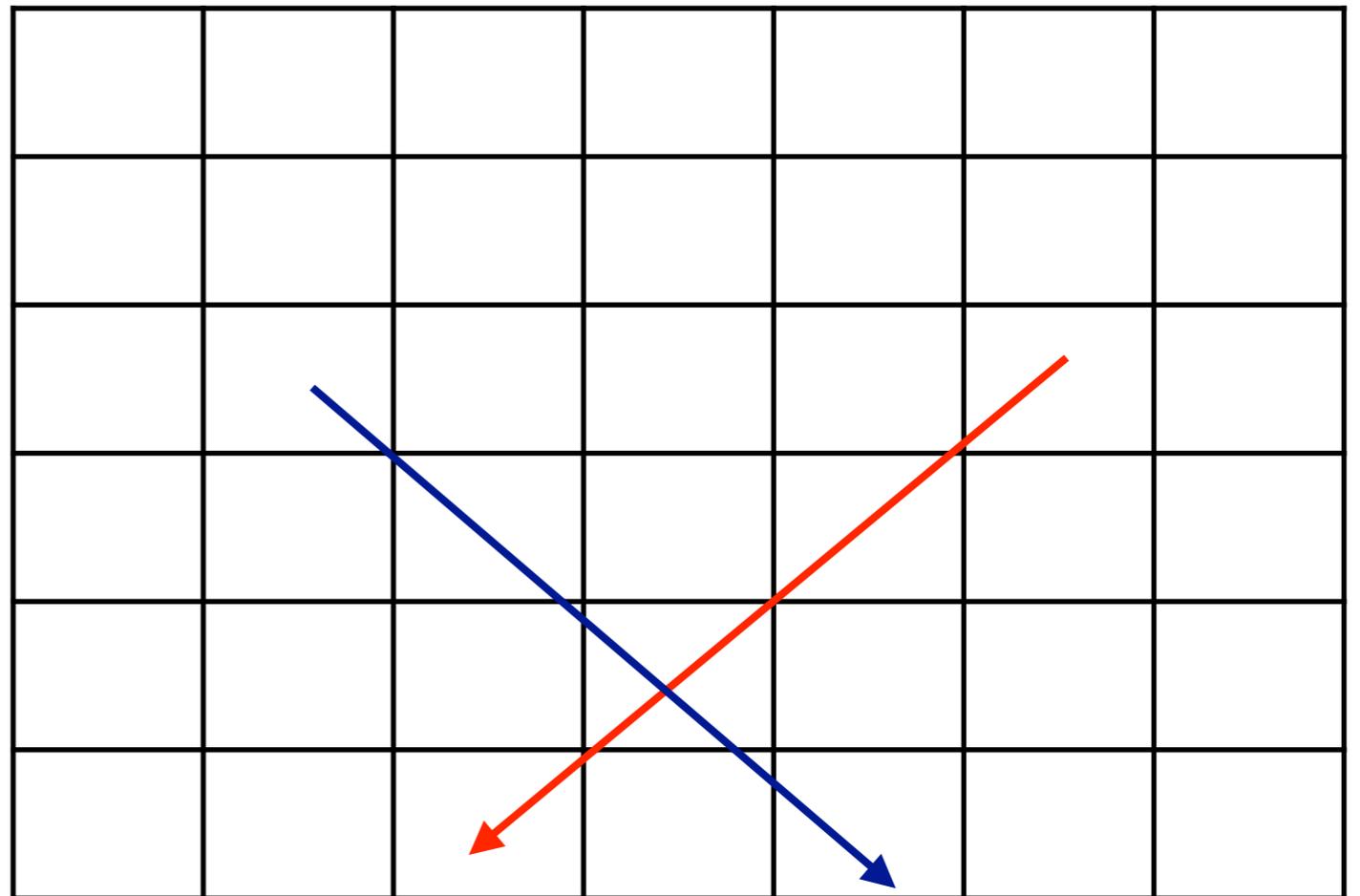
nacci@elet.polimi.it - alessandronacci.com





Controlla vincita obliquo

- In obliquo abbiamo due direzioni possibili
- Quindi creiamo una funzione parametrica che dato un parametro direzione estra un vettore orizzontale...



```
int estra_controlla_vettore_obliquo(int x, int y, char schermo[SCREEN_W][SCREEN_H],  
                                     char occorrenza, int orientamento)
```



Estrazione array obliquo (codice C)

```
int extra_controlla_vettore_obliquo(int x, int y, char schermo[SCREEN_W][SCREEN_H],  
                                     char occorrenza, int orientamento)  
{  
  
    char vettore[DIAG];  
    int curr_x, curr_y;  
    int i = 0;  
  
    do {  
        curr_x = x + (orientamento) * i;  
        curr_y = y + i;  
        vettore[i] = schermo[curr_x][curr_y];  
        i++;  
    } while (curr_x >= 0 && curr_x < SCREEN_W && curr_y >= 0 && curr_y < SCREEN_H);  
  
    return controlla_vettore(vettore, i-1, occorrenza);  
}
```



Controllo vincita in obliquo

- Quindi, usando la funzione appena creata ora possiamo controllare le vincite in obliquo

```
int controlla_obliquo(char schermo[SCREEN_W][SCREEN_H], char occorrenza)
{
    int x,y,i;
    int risultato_controllo = 0;

    for (y = 0; y < SCREEN_H; y++)
        for (x = 0; x < SCREEN_W; x++)
        {
            risultato_controllo = extra_controlla_vettore_obliquo(x,y,schermo,occorrenza, 1);
            if (risultato_controllo) return 1;

            risultato_controllo = extra_controlla_vettore_obliquo(SCREEN_W-x,y,schermo,occorrenza, -1);
            if (risultato_controllo) return 1;
        }

    return 0;
}
```



Controllo di una qualsiasi vincita

- Usando le funzioni precedenti possiamo controllare una qualsiasi vincita

```
int controlla_vincita(char schermo[SCREEN_W][SCREEN_H], char simbolo_pedina)
{
}
}
```



- Prima di scrivere il main(), ci mancano ancora un paio di funzioni comode...

```
// Restituisce una variabile 'giocatore' dato il nome del giocatore e il simbolo della pedina
giocatore crea_giocatore(char nome[MAX_NOME_GIOCATORE], char pedina);

// Richiede a schermo dove inserire pedina. Restituisce '1' se tutto ok, '0' in caso di errore
int richiedi_inserimento_pedina();

// Inserisce la pedina 'pedina' nella colonna 'colonna' dello schermo di gioco 'schermo'
int inserisci_pedina(char schermo[SCREEN_W][SCREEN_H], int colonna, char pedina);
```



Crea giocatore: codice C

```
giocatore crea_giocatore(char nome[MAX_NOME_GIOCATORE], char simbolo_pedina)
{
    giocatore g;
    strcpy(g.nome, nome);
    g.simbolo_pedina = simbolo_pedina;
    return g;
}
```

```
typedef struct {
    char nome[MAX_NOME_GIOCATORE];
    char simbolo_pedina;
} giocatore;
```



Richiedi inserimento pedina: codice C

```
int richiedi_inserimento_pedina()  
{  
    int colonna;  
    printf("In quale colonna vuoi inserire la pedina? ==> ");  
    scanf("%d", &colonna);  
    return colonna;  
}
```



Inserisci pedina: codice C

```
int inserisci_pedina(char schermo[SCREEN_W][SCREEN_H], int colonna, char simbolo_pedina)
{
    int y;
    punto_schermo p;
    forma pedina;

    for (y = SCREEN_H-2; y >= 0; y--)
    {
        if (schermo[colonna][y] == ' ')
        {
            p = crea_punto_schermo(colonna, y, 0);
            pedina = genera_punto(simbolo_pedina);
            disegna_forma(pedina, p, schermo);
            return 1;
        }
    }

    return 0;
}
```



- Ed ora, scriviamo il main:
 - Dobbiamo creare due giocatori
 - Dobbiamo gestire l'alternanza dei due giocatori
 - Dobbiamo richiedere dove inserire la pedina
 - Dobbiamo controllare la vincita

```

int main(){

    char schermo[SCREEN_W][SCREEN_H];
    giocatore giocatori[NUMERO_GIOCATORI];

    int giocatore_corrente = 0;
    int vincitore = -1;
    int colonna_richiesta;

    // Disegniamo un quadrato
    giocatori[0] = crea_giocatore("MARIO", 'o');
    giocatori[1] = crea_giocatore("PAOLO", 'x');

    inizializza_schermo(schermo);
    disegna_schermo(schermo);
    while(vincitore == -1)
    {
        printf ("Giocatore %d \n", giocatore_corrente + 1);
        colonna_richiesta = richiedi_inserimento_pedina();
        if (inserisci_pedina(schermo, colonna_richiesta, giocatori[giocatore_corrente].simbolo_pedina)){
            if (controlla_vincita(schermo, giocatori[giocatore_corrente].simbolo_pedina))
            {
                printf("Il giocatore %d, %s ha vinto!\n\n",
                    giocatore_corrente + 1, giocatori[giocatore_corrente].nome);
                return 0;
            }
            giocatore_corrente = (giocatore_corrente + 1) % NUMERO_GIOCATORI;
        } else {
            printf("La colonna inserita non è valida. Seleziona un'altra colonna.\n");
            aspetta_invio();
            aspetta_invio(); // RISOLVERE QUESTO PROBLEMA
        }
        disegna_schermo(schermo);
    }
}

```

**Tutte il materiale sarà
disponibile sul mio sito internet!**

alessandronacci.com

**Potete lasciare il vostro giudizio
qui:**

<http://tinyurl.com/IEIMExe2014>

See You Next Time!

