23 Ottobre 2023

Programmazione B
Ingegneria e Scienze Informatiche - Cesena
A.A. 2023-2024

Elaborato 6

Data di sottomissione: entro le 23:59 del **5 Novembre 2023**.
Formato di sottomissione: un file compresso con nome `elaborato6.zip`, contenente un unico file sorgente con nome `game.c`

Specifiche:

- Sviluppare funzioni di libreria per poter gestire i movimenti delle racchette e della palla nel gioco *pong*.
- Viene fornita l'implementazione dell'intero gioco, tranne l'implementazione della libreria `game.c`. L'implementazione fa uso della libreria `curses`.
- I prototipi delle funzioni da implementare sono dichiarati nell'header `game.h` e allegati alle specifiche.
- Le funzioni di libreria in `game.c` si occupano di gestire gli aggiornamenti delle posizioni delle racchette e della palla, in maniera consistente con l'ambiente di gioco.
- Gli oggetti `paletta` e `palla` devono essere definiti tramite strutture nella libreria `game.c` e tali strutture non devono essere visibili all'esterno.

Vincoli:

- Le implementazioni devono aderire ai prototipi e alle specifiche fornite.
- Le eventuali funzioni di utility della libreria devono essere *nascoste*.
- Non è possibile utilizzare puntatori o la notazione specifica per i puntatori per lo sviluppo delle funzioni di libreria.

Suggerimenti:

- Iniziare con le implementazioni delle funzioni `get_*` e successivamente con le funzioni `move_pad*`.

- Modificare `TIMEOUT` in `display.h` e `DELAY` in `main.c` per rendere il gioco più o meno veloce.
- Attenzione: la funzione `setup_game()` *resetta* la partita attuale (quindi anche il punteggio).
- Attenzione: la posizione `x` indica una coordinata sull'asse delle ascisse (orizzontale) mentre la posizione `y` indica una coordinata sull'asse delle ordinate (verticale).

Descrizione degli elementi del gioco

- **Tavola di gioco**. Sia le racchette che la palla si spostano in una tavola di gioco (matrice) di dimensione (`height+1`)×(`width+1`), dove la posizione (0,0) corrisponde all'angolo in alto a sinistra e la posizione (`height`,`width`) corrisponde all'angolo in basso a destra.

- **Racchette**. Le racchette sono rappresentate come barre verticali di lunghezza fissa. La lunghezza delle racchette viene scelta opportunamente ad ogni esecuzione a seconda della dimensione della tavola di gioco. Indichiamo genericamente la racchetta più a sinistra nel tavolo di gioco come racchetta 1 e quella più a destra come racchetta 2. Una racchetta di lunghezza n occuperà nella tavola di gioco le posizioni:

  $$(y, x), (y + 1, x), ..., (y + n - 1, x)$$

  dove

  $$0 \leq x \leq \texttt{width} \text{ e } 0 \leq y \leq y + n - 1 \leq \texttt{height}$$

  Una racchetta può spostarsi unicamente in verticale. La posizione x di una racchetta resterà quindi invariata per l'intera durata del gioco, mentre la posizione y potrà variare entro i limiti della tavola. D'ora in poi, quando parliamo di *posizione della racchetta* intendiamo unicamente la posizione (y, x). Lo spazio occupato da una racchetta può essere univocamente determinato da tale posizione e dalla sua lunghezza.

  La posizione iniziale delle due racchette è definita dai parametri `pad1_pos` e `pad2_pos` della funzione `setup_game()`. La lunghezza delle due racchette è definita dal parametro `pad_len` della funzione `setup_game()`.

- **Palla**. Una palla è rappresentata (per questioni grafiche) da due punti:

  $$(y, x) \text{ e } (y, x + 1)$$

  dove

  $$0 \leq x < x + 1 \leq \texttt{width} \text{ e } 0 \leq y \leq \texttt{height}$$

D'ora in poi, quando parliamo di *posizione della palla*, intendiamo unicamente la posizione (y, x). Lo spazio occupato dalla palla può essere univocamente determinato da tale posizione.

La posizione iniziale della palla è definita dal parametro `ball_pos` della funzione `setup_game()`. Ogni volta che uno dei due giocatori segna un punto, la palla deve essere resettata in tale posizione iniziale.

Descrizione dei movimenti nel gioco

- **Movimento delle racchette**. Le racchette possono muoversi unicamente in direzione verticale. Non tutti gli spostamenti verticali sono ammissibili. I movimenti vietati sono descritti di seguito. Assumiamo che una racchetta sia in posizione (y, x) e che abbia lunghezza n su una tavola di dimensione $(\texttt{height}+1)\times(\texttt{width}+1)$.
  Non è possibile spostare la racchetta in alto di una posizione se:

  1. Siamo al limite della tavola di gioco, i.e. $y = 0$, oppure
  2. la palla tocca il bordo superiore della racchetta (vedi Fig. 2).

  Non è possibile spostare la racchetta in basso di una posizione se:

  1. Siamo al limite della tavola di gioco, i.e. $y + n - 1 = \texttt{height}$, oppure
  2. la palla tocca il bordo inferiore della racchetta (vedi Fig. 3).

- **Movimento della palla**. La palla può muoversi in direzione orizzontale, `LEFT` e `RIGHT`, e verticale, `UP` e `DOWN`, codificate nel seguente modo:

  $$\texttt{LEFT} = -1,\ \texttt{RIGHT} = 1,\ \texttt{UP} = -1,\ \texttt{DOWN} = 1$$

  La direzione corrente verso cui viaggia la palla è una coppia di valori (V, H) dove:

  $$V \in \{\texttt{UP}, \texttt{DOWN}\} \text{ e } H \in \{\texttt{LEFT}, \texttt{RIGHT}\}$$

  Se la palla è attualmente nella posizione (y, x) e viaggia in direzione (V, H), allora la nuova posizione sarà semplicemente $(y + V, x + H)$. Quando richiesto nel gioco (vedi funzione `move_ball()`) dovremo aggiornare la direzione di movimento della palla e calcolarne la nuova posizione.

  La direzione di movimento della palla cambierà spesso durante l'esecuzione del gioco. Nel movimento devono sempre essere valide le seguenti condizioni. Assumiamo che la palla sia in posizione (y, x) su una tavola di dimensione $\texttt{height}\times\texttt{width}$.

  (a) Se la palla tocca il bordo alto della tavola di gioco, i.e. $y = 0$, la direzione sull'asse verticale deve essere **sempre** settata a `DOWN`. La direzione sull'asse orizzontale resta invariata a meno che questo non violi il caso (e).

(b) Se la palla tocca il bordo basso della tavola di gioco, i.e. $y =$ `height`, la direzione sull'asse verticale deve essere **sempre** settata ad `UP`. La direzione sull'asse orizzontale resta invariata a meno che questo non violi il caso (e).

(c) Se la palla tocca il bordo sinistro della tavola di gioco, i.e. $x = 0$, il giocatore 2 segna un punto e si ricomincia con un nuovo round. Viene modificata la posizione della palla, che torna alla posizione iniziale. La direzione attuale non viene modificata.

(d) Se la palla tocca il bordo destro della tavola di gioco, i.e. $x+1 =$`width`, il giocatore 1 segna un punto e si ricomincia con un nuovo round. Viene modificata la posizione della palla, che torna alla posizione iniziale. La direzione attuale non viene modificata.

(e) Se la palla tocca una racchetta, la direzione orizzontale viene **sempre** modificata secondo le seguenti regole. Se la palla tocca la racchetta 1, la direzione sull'asse orizzontale deve essere **sempre** settata a `RIGHT`. Se la palla tocca la racchetta 2, la direzione sull'asse orizzontale deve essere **sempre** settata a `LEFT`. Abbiamo tre categorie di tocco (vedi le Figure 1-3) che determinano invece la direzione verticale:

   1. *Tocco pieno* (Fig 1). La direzione sull'asse verticale resta invariata a meno che questo non violi i casi (a) e (b).
   2. *Tocco sul bordo superiore* (Fig 2). La direzione sull'asse verticale deve essere settata ad `UP`, a meno che questo non violi il caso (a).
   3. *Tocco sul bordo inferiore* (Fig 3). La direzione sull'asse verticale deve essere settata a `DOWN`, a meno che questo non violi il caso (b).

(f) Se non si applicano nessuno dei casi precedenti, la direzione di movimento non viene modificata.

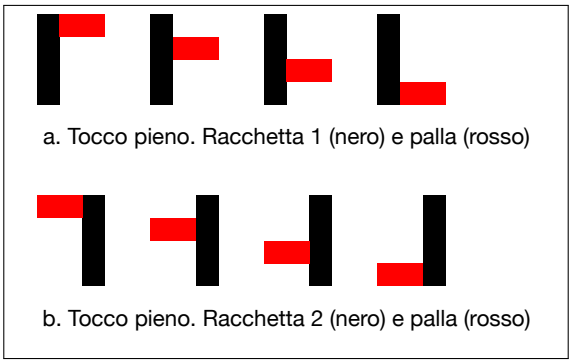

Figura 1: Tocco pieno.

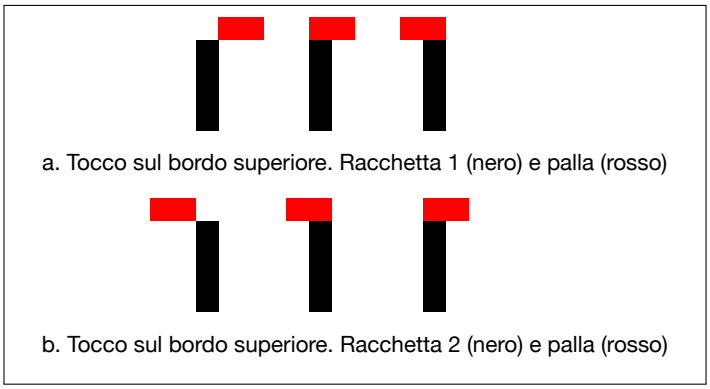

Figura 2: Tocco sul bordo superiore.

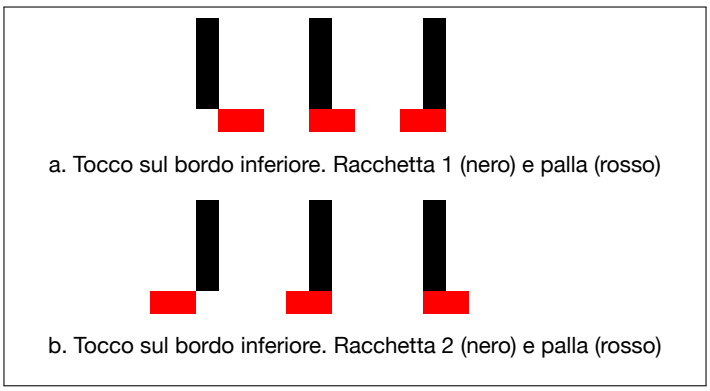

Figura 3: Tocco sul bordo inferiore.

Descrizione dettagliata delle funzioni

- `setup_game()`: setta la configurazione iniziale del gioco. In dettaglio:
  - Le dimensioni della tavola di gioco sono fissate per l'intera durata del gioco dai due argomenti `height` e `width`.
  - La posizione iniziale della palla è specificata dai punti

    $$0 \leq x \leq \texttt{width} \text{ e } 0 \leq y \leq \texttt{height}$$

    dove `x` e `y` sono i campi della struttura `ball_pos`. Ogni volta che uno dei due giocatori segna un punto, la palla deve essere riposizionata in tale posizione iniziale.
  - La direzione di movimento iniziale della palla è specificata dalle direzioni

    $$\texttt{x} \in \{-1, 1\} \text{ e } \texttt{y} \in \{-1, 1\}$$

    dove `x` (direzione orizzontale) e `y` (direzione verticale) sono i campi della struttura `ball_dir`.
  - La posizioni iniziali delle racchette 1 e 2 sono specificate rispettivamente dai punti

    $$0 \leq x \leq \texttt{width} \text{ e } 0 \leq y \leq \texttt{height}$$

    dove `x` e `y` sono campi della struttura `pad1_pos` e `pad2_pos`, rispettivamente per le due racchette.
  - La lunghezza delle due racchette è specificata dall'argomento `pad_len`.
- `move_ball()`: aggiorna **prima** la direzione di movimento della palla (se necessario) e **sucessivamente** ne effettua lo spostamento (vedi **movimento palla**). In caso di punteggio da parte di uno dei due giocatori, aggiorna lo score corrente e sposta la palla nella posizione iniziale. In ogni caso, la palla deve essere spostata una sola volta durante la chiamata e lo spostamento deve essere l'ultima operazione effettuata dalla funzione.
- `move_pad1_up()`: sposta la racchetta 1 di una posizione in alto. Se non è possibile spostare la racchetta in alto (vedi **movimento racchette**), la posizione attuale resta invariata.
- `move_pad1_down()`: sposta la racchetta 1 di una posizione in basso. Se non è possibile spostare la racchetta in basso (vedi **movimento racchette**), la posizione attuale resta invariata.
- `move_pad2_up()`: sposta la racchetta 2 di una posizione in alto. Viene gestita come per la racchetta 1.
- `move_pad2_down()`: sposta la racchetta 2 di una posizione in basso. Viene gestita come per la racchetta 1.
- `get_pad1_pos()` ritorna la posizione corrente della racchetta 1.
- `get_pad2_pos()`: ritorna la posizione corrente della racchetta 2.
- `get_pad1_len()` ritorna la lunghezza della racchetta 1.
- `get_pad2_len()`: ritorna la lunghezza della racchetta 2.
- `get_pad1_score()` ritorna il punteggio corrente del giocatore 1.
- `get_pad2_score()`: ritorna il punteggio corrente del giocatore 2.

```c
#ifndef GAME_H
#define GAME_H

struct position {
 int x;
 int y;
};

/*
 * Setup a game with the following starting configuration:
 * - table dimension equal to heigth*width
 * - ball starting position at ball_pos
 * - ball starting direction towards ball_dir
 * - pad1 starting position at pad1_pos
 * - pad2 starting position at pad2_pos
 * - pad length equal to pad_len
 */
void setup_game(int height, int width,
                struct position ball_pos, struct position ball_dir,
                struct position pad1_pos, struct position pad2_pos, int pad_len);

/* Moves pad1 one position up. */
void move_pad1_up(void);

/* Moves pad2 one poisiton up */
void move_pad2_up(void);

/* Moves pad1 one position down. */
void move_pad1_down(void);

/* Moves pad2 one position down. */
void move_pad2_down(void);

/* Moves the ball in the current direction */
void move_ball(void);

/* Returns ball current position */
struct position get_ball_pos(void);

/* Returns pad1 current position */
struct position get_pad1_pos(void);

/* Returns pad2 current position */
struct position get_pad2_pos(void);

/* Returns the pad length */
unsigned int get_pad_len(void);

/* Returns pad1 current score */
unsigned int get_pad1_score(void);

/* Returns pad2 current score */
unsigned int get_pad2_score(void);

#endif
```