

14 Novembre 2022

Programmazione B
Ingegneria e Scienze Informatiche - Cesena
A.A. 2022-2023

Elaborato 8

Data di sottomissione: entro le 20 del 27 Novembre 2022.

Formato di sottomissione: un file compresso con nome `elaborato8.zip`, contenente un unico file sorgente con nome `ghosts.c`

Codeboard: <https://codeboard.io/projects/131851/>

Specifiche:

- Sviluppare funzioni di libreria per poter gestire alcuni elementi nel gioco `pacman`.
- Viene fornita l'implementazione dell'intero gioco, tranne l'implementazione della libreria `ghosts.c`.
- L'implementazione fa uso della libreria `curses`, non necessaria per l'implementazione di `ghosts.c`.
- I prototipi delle funzioni da implementare sono dichiarati nell'header `ghosts.h`, allegato alle specifiche.
- La struttura `struct ghosts` è dichiarata in modo opaco nell'header `ghosts.h`, deve quindi essere definita nel file `ghosts.c`

Vincoli:

- Le implementazioni devono aderire perfettamente ai prototipi e alle specifiche fornite.
- Le eventuali funzioni di utility della libreria e variabili globali devono essere *nascoste* all'esterno.

Descrizione dettagliata delle funzioni:

- `ghosts_setup()`: crea la struttura dati ideata per memorizzare le informazioni di `num_ghosts` fantasmi e ne ritorna un puntatore.
- `ghosts_destroy()`: libera tutta la memoria allocata dinamicamente per la creazione della `struct ghosts`.
- `ghosts_set_arena()`: memorizza la matrice di gioco che indica le posizioni in cui i fantasmi potranno muoversi (vedi sotto).
- `ghosts_set_position()`: setta la posizione per il fantasma con identificativo `id`.
- `ghosts_set_status()`: setta lo stato (`NORMAL`, `SCARED_NORMAL`, `SCARED_BLINKING`, `EYES`) del fantasma con identificativo `id`.
- `ghosts_get_number()`: ritorna il numero totale di fantasmi.
- `ghosts_get_position()`: ritorna la posizione nell'arena del fantasma con identificativo `id`.
- `ghosts_get_status()`: ritorna lo stato del fantasma con identificativo `id`.
- `ghosts_move()`: sposta il fantasma con identificativo `id` secondo i seguenti vincoli:
 - E' possibile effettuare un movimento unicamente nelle direzioni:
 - * `UP` (la cella immediatamente sopra la posizione corrente),
 - * `DOWN` (la cella immediatamente sotto),
 - * `LEFT` (la cella immediatamente a sinistra)
 - * `RIGHT` (la cella immediatamente a destra).
 - Se lo stato del fantasma è `NORMAL`, lo spostamento in una cella può essere effettuato solo se tale cella è libera:
 - * la cella non è occupata da un altro fantasma
 - * la cella non è occupata da un muro
 - Se lo stato del fantasma non è `NORMAL`, lo spostamento non può essere effettuato in una cella occupata da un muro, da un altro fantasma e da pacman.
 - Se il bordo non contiene un muro è possibile attraversare l'arena di gioco e sbucare dalla parte opposta.

- Deve sempre essere effettuato un movimento, tranne nel caso in cui nessuna delle quattro posizioni (`UP`, `DOWN`, `LEFT`, `RIGHT`) sia libera secondo le regole descritte sopra.

Il movimento di un fantasma può essere random (gioco noioso), a patto che rispetti i vincoli definiti sopra. L'obiettivo dovrebbe essere quello di cercare di implementare un movimento non banale, seguendo i seguenti principi:

- se lo stato del fantasma è `NORMAL` allora il movimento deve cercare di portarlo vicino a pacman;
- se lo stato del fantasma è `SCARED_NORMAL` oppure `SCARED_BLINKING` allora il movimento deve cercare di portarlo lontano da pacman;
- se lo stato del fantasma è `EYES` allora il movimento deve cercare di riportarlo in una delle posizioni `HOME` (le posizioni nella matrice marcate con `X`) in modo che possa riprendere la forma normale. In quest'ultimo caso è sufficiente scegliere la direzione di movimento *suggerita* nella posizione corrente della matrice `arena`.

Descrizione dell'`arena` di gioco passata alla funzione `ghosts_set_arena()`:

- L'`arena` viene passata come matrice di caratteri.
- Tale matrice è ricavata dal file `positions.txt` e *processata* in `arena.c`: vengono rimossi i simboli corrispondenti a pacman e fantasmi e vengono calcolati i percorsi per ritornare nella home.
- Ogni cella contiene un carattere che indica se tale cella è *attraversabile* oppure no (contiene un muro).
- Ogni cella attraversabile contiene una lettera che indica la direzione da seguire per poter ritornare velocemente nella casa dei fantasmi (vedi file `path.txt` nel pacchetto sorgente):
 - Walls: `x` (`XWALL_SYM`)
 - Home: `X` (`HOME_SYM`)
 - Up: `U` (`UP_SYM`)
 - Left: `L` (`LEFT_SYM`)
 - Right: `R` (`RIGHT_SYM`)
 - Down: `D` (`DOWN_SYM`)

```
1 #ifndef GHOSTS_H
2 #define GHOSTS_H
3
4 #include "global.h"
5 #include "pacman.h"
6
7 struct ghosts; /* Opaque declation of the ghosts data structure */
8
9 /* Create the ghosts data structure */
10 struct ghosts *ghosts_setup(unsigned int num_ghosts);
11
12 /* Destroy the ghost data structure */
13 void ghosts_destroy(struct ghosts *G);
14
15 /* Set the arena (A) matrix */
16 void ghosts_set_arena(struct ghosts *G, char **A, unsigned int nrow,
17                                     unsigned int ncol);
18
19 /* Set the position of the ghost id. */
20 void ghosts_set_position(struct ghosts *G, unsigned int id, struct position pos);
21
22 /* Set the status of the ghost id. */
23 void ghosts_set_status(struct ghosts *G, unsigned int id, enum ghost_status status);
24
25 /* Return the number of ghosts */
26 unsigned int ghosts_get_number(struct ghosts *G);
27
28 /* Return the position of the ghost id. */
29 struct position ghosts_get_position(struct ghosts *G, unsigned int id);
30
31 /* Return the status of the ghost id. */
32 enum ghost_status ghosts_get_status(struct ghosts *G, unsigned int id);
33
34 /* Move the ghost id (according to its status). Returns the new position */
35 struct position ghosts_move(struct ghosts *G, struct pacman *P, unsigned int id);
36
37 #endif
```