



Extra Points Part 2

Giovanni Pollo (290136)

1 Descrizione Funzioni

Il codice è stato organizzato utilizzando una libreria (*gioco.h*) con relativo file *gioco.c* che contiene le implementazioni delle funzioni. La scelta di utilizzare un file dedicato alle principali funzioni del progetto è stata fatta per rendere il codice il più ordinato possibile.

In particolare, nel file *gioco.c*, sono presenti le seguenti funzioni:

- **Joystick:** Per la gestione del joystick vengono utilizzate le funzioni *joystick_select* (per il click del joystick), *joystick_nord* (per la posizione verso l'alto), *joystick_sud* (per la posizione verso il basso), *joystick_est* (per la posizione verso destra) e *joystick_ouest* (per la posizione verso sinistra).
- **Draw:** Per i disegni sono utilizzate le funzioni *draw_arrow* (per disegnare la freccia del player), *draw_player* (per disegnare il player in base alla posizione in cui si trova) e *draw_obstacle* (per disegnare gli ostacoli). Le ultime due, si basano su un metodo chiamato *LCD_DrawRectangle*, che è stato dichiarato nel file *GLCD.h* ed implementato all'interno del file *GLCD.c*.
- **Movimento:** Per muovere il giocatore, in base alla direzione in cui si trova, è stata fatta una funzione dedicata, chiamata *muovi*. Per evitare movimenti nel caso in cui ci si trovi attaccati ai margini del labirinto, si è scelto di utilizzare un semplice *if* che controlli preventivamente la posizione del player.
- **Distanza:** Anche per il calcolo della distanza si utilizza una funzione dedicata, in particolare *calcola_distanza*.
- **Vittoria:** Per la vittoria si utilizza la funzione *win* che ha il compito di disabilitare il *RIT*, grazie alla *disable_RIT()*, così da interdire il funzionamento del joystick e di disegnare la scritta di vittoria "YOU WON".

2 Timer e RIT

Il RIT è stato utilizzato per il joystick. In particolare, lo si è inizializzato con un valore tale da consentire un timing di $50ms$. Di conseguenza, per gestire il movimento del player dopo 1 secondo, si è scelto attendere che la variabile interna al joystick raggiungesse 20 (infatti $1s = 50ms \cdot 20$).

Per gestire la pressione del display (touch), la scelta è ricaduta sul timer 0. In particolare quest'ultimo è stato inizializzato al valore $0x4E2$, che corrisponde a $50\mu s$. All'interno dell'interrupt del timer (*TIMER0_IRQHANDLER*) sono stati implementati i metodi per il riconoscimento della pressione del display con le relative azioni da eseguire.

3 Touch Panel

Per riconoscere il tocco del touch panel, si è utilizzata la *get_DisplayPoint* all'interno della routine di interrupt del timer 0. Questo ha permesso di implementare un meccanismo di Polling a cadenza $50\mu s$. Per verificare dove fosse stato premuto il display, è bastato andare a controllare i valori di *display.x* e *display.y*.