Immagine che contiene quadrato, Rettangolo, schermata, testo

Descrizione generata automaticamenteQUORIDOR su LANDTIGER **APPLICATION NOTE**

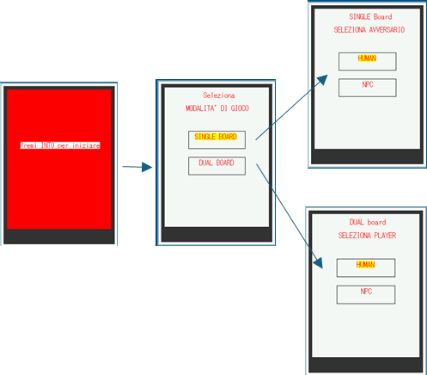
Carlo MIGLIACCIO s332937

*Obiettivo del presente documento è quello di spiegare brevemente – dando qualche dettaglio tecnico – le caratteristiche aggiunte all’implementazione del gioco* ***Quoridor per la scheda LandTiger****. Tale release, in particolare, amplia e rivede le feature di una “versione base” precedentemente sviluppata.*

1. **MENU**

Permette di far scegliere all’utente:

* La *modalità di gioco* tra *SINGLE* e *MULTIPLAYER;*
* Il *tipo di giocatore* tra *HUMAN* e *NPC* (par. 3)



2

1

INT0



JOYSTICK

Le subroutine adibite alla gestione del menù, implementate sfruttando la libreria GLCD.h, sono:



2

* **void** VisualizzaMenu(QualeMenu quale) cambia, in base alla variabile quale, il tipo di schermata visualizzata: selezione *modalità di gioco* (1 in figura) o selezione *tipo giocatore* (2 in figura).
* **void** CambiaEvidenziazione(**int** Cosa, **int** Quale) usata insieme ai tasti direzionali del joystick per evidenziare la scelta dell’utente.

Dal momento che parte dei tasti del joystick sono utilizzati anche per scegliere le mosse, per differenziare le azioni da svolgere è stata introdotta una **variabile di stato** di tipo STATO che potesse monitorare l’attività del gioco (si parte in modalità MODE).

**1 typedef** **enum** {MODE, PLAYER\_SEL, PLAY} STATO;

**2** STATO StatoGioco=MODE;

In particolare, le routine di gestione dei tasti UP, DOWN, SEL usano queste informazioni per decidere (tramite **switch-case**) quale compito indirizzare.

Le scelte (al passo 1 e 2) vengono invece salvate nelle variabili game\_mode (1) e opposite/mine (2) (a seconda della modalità scelta) e usate in modo opportuno dalle routine di gestione del gioco. Valore di init=-1.

**1** **typedef** **enum** {SINGLE, DUAL} GAME\_MODE;

**2** **typedef** **enum** {HUMAN, NPC} PLAYER\_TYPE;

**3** GAME\_MODE game\_mode=-1;

**4** PLAYER\_TYPE opposite=-1;

**5** PLAYER\_TYPE mine=-1;

1. **AGGIUNTA NUOVE MOSSE**

*Nota: L’aggiunta di queste nuove mosse completa il set di regole del gioco originale [[1]](#footnote-1), solo parzialmente coperto nella versione precedente.*

|  |  |
| --- | --- |
| **Faccia a faccia** | **Faccia a faccia +muro dietro** |
| *Immagine che contiene quadrato, Rettangolo, schermata, testo  Descrizione generata automaticamente Immagine che contiene quadrato, Rettangolo, schermata  Descrizione generata automaticamente* |  |

Per poter supportare queste nuove mosse:

* È stata irrobustita la funzione Check(), che durante ogni turno evidenzia le mosse possibili, aggiungendo **nuove flag** (UPRIGHT, UPLEFT, DOWNRIGHT, DOWNLEFT) e controlli che limitano o garantiscono il movimento del token;
* Viene data la possibilità all’utente di muoversi con il joystick anche nelle **quattro direzioni diagonali** (quando consentito). A tale scopo l’ISR del RIT (timer che si occupa della gestione in polling del joystick) contempla i casi in cui due direzioni siano attivate contemporaneamente.
* Sono state aggiunte quindi le seguenti condizioni di verifica sui pin (attivo bassi) della PORT1 associati alla periferica (vedi Schematici LandTiger[[2]](#footnote-2))

|  |  |
| --- | --- |
| **Nuova direzione** | **Condizioni da verificare** (IRQ\_RIT.c) [[3]](#footnote-3) |
| UP\_RIGHT | (LPC\_GPIO1->FIOPIN &(1<<**29**))== 0 && (LPC\_GPIO1->FIOPIN & (1<<**28**)) == 0) |
| UP\_LEFT | (LPC\_GPIO1->FIOPIN &(1<<**26**))== 0 && (LPC\_GPIO1->FIOPIN & (1<<**27**)) == 0) |
| DOWN\_RIGHT | (LPC\_GPIO1->FIOPIN &(1<<**26**))== 0 && (LPC\_GPIO1->FIOPIN & (1<<**28**)) == 0) |
| DOWN\_LEFT | (LPC\_GPIO1->FIOPIN &(1<<**26**))== 0 && (LPC\_GPIO1->FIOPIN & (1<<**27**)) == 0) |

Come ci si può aspettare sono state introdotte altrettante variabili di stato che gestiscono il polling delle quattro nuove direzioni.

**Problema riscontrato:** durante il test delle nuove funzionalità, si è notato che non sempre su tutte le schede si riesce a muovere facilmente il joystick nelle direzioni diagonali.

1. **NPC (Player automatico)**

È la situazione in cui si sceglie come *opponent* (opposite=NPC, game\_mode=SINGLE) o *proprio giocatore* (mine=NPC, game\_mode=DUAL) NPC; in questo contesto, la mossa in uno dei due turni viene fatta in modo automatico.

La funzioni preposte a tale compito sono:

* **void** MossaNPC(\_typePLAYER\* whoPt) genera in modo *pseudocasuale*uno tra i due tipi di mossa possibili (MOVE, WALL). A questo punto:
  + Se Modalità MOVE 🡪 Scegli Direzione
  + Se Modalità WALL 🡪 Scegli Centro Muro, Scegli Orientamento
* **void** MossaNPC\_wrapper(**void**) è una “funzione-involucro” che si occupa del cambio di turno, invoca MossaNPC(), controlla la vittoria e infine azzera il Timer per far giocare l’opponent.

1. **MULTIPLAYER MODE**

Selezionata come modalità DUAL BOARD, le due schede prima che cominci il gioco si scambiano un **acknowledge** nel modo seguente:

1

ACK\_TRASMESSO: 0xFF000000

Immagine che contiene testo, elettronica, Ingegneria elettronica, circuito

Descrizione generata automaticamente Immagine che contiene testo, elettronica, Ingegneria elettronica, circuito

Descrizione generata automaticamente

2

2

1

ACK\_TRASMESSO: 0xFF000000

ACK\_RICEVUTO:

0xFF000000

ACK\_RICEVUTO:

0xFF000000

Si aspettano 3s per rilevare l’ACK dell’altra scheda (viene usato a tale scopo il TIMER1 attivato dopo il SEL del joystick), si entra quindi nella modalità di scelta player, comincia infine la partita.

Il gioco prosegue come di consueto, con qualche accorgimento:

* Trasmissione mossa: Quando viene fatta una mossa la si trasmette sul bus CAN [[4]](#footnote-4) utilizzando il formato definito nelle specifiche (se ne occupa la funzione **void** TrasmettiCAN\_Msg(**int** Mossa) che usa CAN\_wrMsg() di CAN.c)
* Ricezione Mossa: Il cambio di turno, oltre che dallo scadere del timer (TIMER0), viene gestito dall’ISR del CAN in cui:
  1. Leggendo il messaggio ricevuto, si esegue l’unpack della Mossa ricevuta e la si visualizza;
  2. Si ripristina il turno e nel caso in cui mine=NPC si esegue la mossa in automatico.

1. 🔗 [How to play Quoridor | Official Rules | UltraBoardGames](https://www.ultraboardgames.com/quoridor/game-rules.php#:~:text=Quoridor%20Game%20Rules%201%20Components%20one%20board%2020,Face%20...%207%20End%20of%20the%20Game%20) [↑](#footnote-ref-1)
2. 🔗 <https://www.yumpu.com/en/document/read/42990366/schematic-pdf-mbed> [↑](#footnote-ref-2)
3. 🔗 <https://www.pololu.com/file/0J431/user.manual.lpc17xx.pdf> [↑](#footnote-ref-3)
4. 🔗 <https://it.wikipedia.org/wiki/Controller_Area_Network> [↑](#footnote-ref-4)