

“MISSIONE ARE”

Un gioco spaziale costruito con Arduino

di Emanuele Concas

Un progetto che a prima vista può essere visto come un qualcosa di semplice, ma che all'interno ha una forte e complicata base software. Questo progetto dimostra come con un hardware basilare si possa, attraverso quasi il solo software, creare un piccolo gioco attraverso l'uso sapiente di uno schermo LCD.

IDEA DI BASE

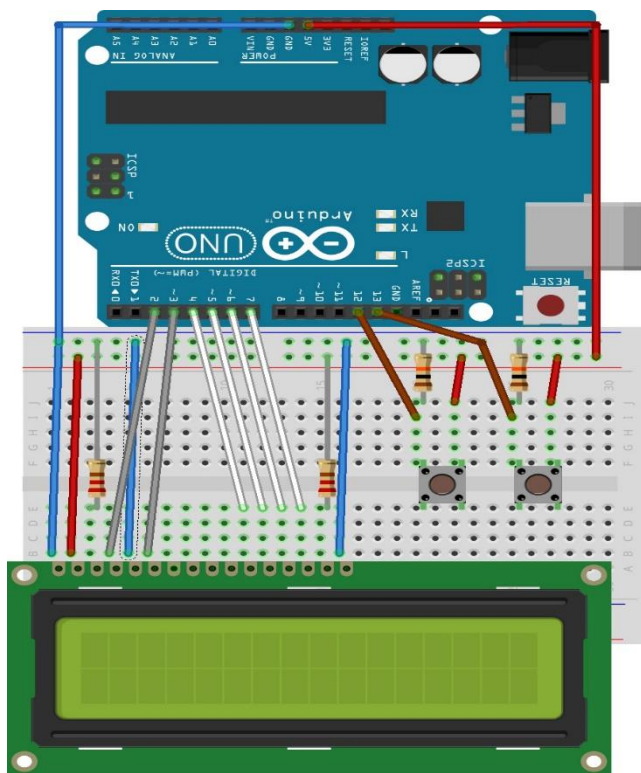
Questo progetto è creato con l'idea di dimostrare l'importanza dell'emulazione e quindi del software, con relativamente pochi componenti è stato possibile ricreare un gioco simil anni 80. Con una logica simile a quella utilizzata in questo progetto è possibile ricreare un qualunque gioco che possa essere rappresentato su uno schermo LCD, da *Pac-Man* a *Donkey Kong*. La reale limitazione è soltanto la grandezza dello schermo, e quindi del mondo di gioco.

Il funzionamento del gioco è concettualmente semplice, la nostra navicella, in posizione centrale, dovrà cercare di difendersi dall'attacco di navicelle aliene, che arriveranno in quantità variabile e con velocità incrementale in base al livello raggiunto. Questo livello è incrementato a 15, 25 e 35 uccisioni, una uccisione equivale ad una navicella abbattuta. Il totale delle navicelle abbattute è visualizzato ed aggiornato nell'angolo destro dello schermo. Le navicelle possono arrivare sia da destra che da sinistra, il nostro shuttle ha infatti due comandi, uno per girarsi e sparare a sinistra ed uno per effettuare l'azione opposta. Il giocatore può riiniziare la partita quando la sua navicella viene distrutta o quando la termina. In questi casi viene mostrato il punteggio finale, che è calcolato in base alle uccisioni, alla distanza percorsa e ad i missili sparati. Così da premiare l'utente che usa meno missili possibili.

Grazie all'uso di define e variabili, è possibile adattare e modificare il gioco in qualunque modo possibile, per esempio si potrebbe aggiungere un sistema di vite al giocatore o modificare il comportamento delle navicelle nemiche con molta semplicità.

Il progetto ha un totale di più di 500 righe di codice, scritte da 0, che permettono di gestire le collisioni tra oggetti, gestire i punteggi e livelli, gestire il movimento dei missili e delle navicelle nemiche.

CIRCUITO E COMPONENTI



Qtà.	Componente	Prezzo
1	Arduino uno	20€
1	LCD screen	5€
2	Resistenza da 220Ω	<1€
2	Resistenza da 10kΩ	<1€
2	Pushbutton	<1€
1	Breadboard	5€

I componenti hardware sono tutti molto economici, le spese maggiori sono per l'acquisto del microcontrollore Arduino, gli altri costi sono pressoché nulli.

L'assemblaggio è molto semplice, lo schermo LCD è stato collegato più o meno come è norma fare, i passaggi sono spiegati sul sito di Arduino¹. È stato inserito un resistore da 220ohm tra il piedino 3 del LCD e massa per mantenere il contrasto costante. I pushbutton sono invece collegati agli ingressi digitali per poter controllare quando e se sono premuti.

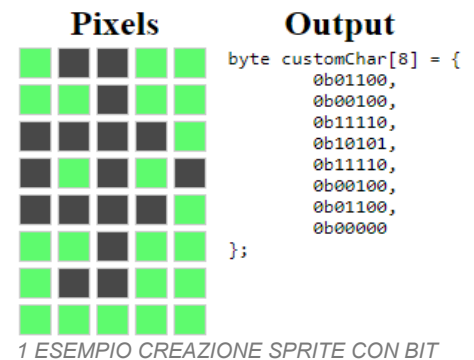
¹ <https://www.arduino.cc/en/Tutorial/HelloWorld>

SOFTWARE

Come anticipato all'inizio, il software è la parte principale e più importante di questo progetto, per facilitare la lettura e la comprensione del codice, sono stati utilizzati diversi commenti lungo il codice. Il sorgente del progetto è disponibile sulla mia pagina *GitHub*².

Le funzioni principali del programma sono le seguenti:

- *updateArea*: è la parte più importante di tutto il progetto, si occupa di aggiornare lo schermo a destra ed a sinistra della nostra navicella, per permettere così il movimento degli spari e degli sprite delle navicelle nemiche.
Qualunque descrizione sarebbe riduttiva per capirne il funzionamento, è quindi consigliato andare a vedere il codice commentato, situato nella *nota 2* a fondo pagina.
- *drawLCD*: all'interno di questa funzione sono dichiarati i disegni che costituiscono i personaggi e le animazioni del nostro gioco. La rappresentazione è in binario, per ogni riga e colonna vale la regola che se la posizione contiene un '1', il led identificato dalla riga e colonna, sarà acceso, viceversa sarà spento. Successivamente avviene lo spostamento degli sprite nella RAM dello schermo LCD e lo schermo viene azzerato.
- *drawShuttle*: aggiorna lo sprite da usare, in base alla direzione ed al momento. Effettua un controllo importantissimo, ovvero quello sulla collisione tra un oggetto e la nostra navicella, elemento fondamentale per capire se si è stati abbattuti. Inoltre si occupa della stampa del numero livello e del numero delle uccisioni.
- *newLevel*: si occupa di controllare le uccisioni, ed eventualmente di terminare il gioco o salire di livello.
- *shuttleFire*: gestisce i missili del nostro shuttle, in base all'orientamento della navicella, sparerà il missile nella direzione esatta.
- *setup*: questa funzione è richiamata soltanto all'avvio di Arduino e contiene l'inizializzazione dei pushbutton, dello schermo LCD e del collegamento seriale, usato per fare il debug.
- *loop*: questa funzione invece viene richiamata continuamente durante l'utilizzo del nostro gioco e contiene al suo interno la chiamata a tutte le funzioni sopra citate, permettendo di gestire le diverse schermate del gioco.



² <https://github.com/conema/missioneAre/blob/master/missioneAre.ino>