

Programmazione di Sistemi Embedded
Jimmy Challenge Arduino Game

Edoardo Rosa - Matr. 707922
Federico Torsello - Matr. 702619

11 giugno 2016

Sommario

In questa relazione si descriverà il gioco Jimmy Challenge. Questo gioco è stato realizzato utilizzando molti componenti hardware/software; sempre con l'obiettivo di sfruttare ed ottimizzare quando più possibile la board Arduino in un'ottica IoT.

Capitolo 1 .

Capitolo 1

Introduzione

Jimmy Challenge è un gioco interattivo in cui un "ladro" tenta di *forzare* un lucchetto utilizzando un *grimaldello* [jimmy in inglese].

Per poter giocare bisogna alimentare l'Arduino e attendere qualche secondo di setup.

Una volta che il settaggio è completo, l'utente può interagire con Arduino utilizzando diversi componenti hardware che mutano il loro comportamento in base al contesto.

1.1 Giocare

L'obiettivo del giocatore è trovare e quindi scassinare il lucchetto nel minor tempo possibile. Questo gioco è giocabile online (uno contro uno) che offline.

La posizione del lucchetto viene assegnata in modo random ad ogni nuovo livello e rimane fissa fino al suo superamento.

Per trovare la posizione attuale del lucchetto bisogna muovere la mano orizzontalmente in direzione del sensore ad ultrasuoni.

Durante le varie fasi di gioco l'utente ha la possibilità di rendersi conto dell'evoluzione del gioco ascoltando i suoni emessi dal buzzer o guardando i colori dei LED.

1.1.1 Significato dei suoni e dei colori

- All'avvio del gioco
 - i LED a 12 pin giallo e rosso fanno un carosello;
 - il LED verde e il buzzer si comportano come quando il lucchetto non è stato trovato.
- Quando **non** si è trovato il lucchetto:
 - il LED verde emette una luce pulsante;

- il LED RGB emette una luce continua di color blu chiaro;
- il buzzer suona due note in modo frenetico.
- Quando si è nell'area del lucchetto:
 - il LED verde emette una luce fissa;
 - il LED RGB continuerà ad emettere una luce continua blu chiaro, ma solo fino a quando non entrerà nello stato di scasso;
 - il buzzer suona due note meno freneticamente.

1.1.2 Superare un livello

Per superare il livello il ladro deve forzare il lucchetto.

Dal punto di vista del giocatore il lucchetto è un'area nello spazio posta davanti al sensore (in linea orizzontale).

Per forzare il lucchetto è sufficiente utilizzare una mano posta davanti al sensore ad ultrasuoni per un tempo delimitato, avviando lo stato di scasso. Se il tempo di scasso non viene rispettato o la mano viene rimossa troppo presto, il livello riparte senza salvare i progressi.

- **Non** si supera il livello:
 - Se la mano viene spostata dall'area del lucchetto troppo presto, per esempio non si è ancora nello stato di scasso
 - Se si rimane troppo tempo nella fase di scasso (il lucchetto è stato "rotto").
- Si può superare il livello:
 - Se la mano resta fissa nella posizione in cui si trova il lucchetto, rispettando il tempo nello **stato di scasso** e poi la si agita sempre nell'area del lucchetto ("aprendolo").

1.1.3 Stato di scasso

Se si rispetta il tempo nello stato di scasso e quindi si apre il lucchetto, si supera il livello.

Per indicare lo stato di scasso si è utilizzato il LED RGB.

- Ogni colore ha un significato:
 - blu scuro: si sta scassinando il lucchetto
 - verde: il lucchetto è scassinato.
- NB: per passare al livello successivo si deve togliere la mano e riposizionarla nell'area del lucchetto [come se si infilasse la "*chiave*".]

- arancio: attenzione, se non si toglie la mano ora si rischia di rompere il lucchetto
- giallo: pericolo di rottura ancora più elevato
- rosso: il lucchetto è stato rotto, quindi il livello deve essere ricominciato di nuovo.

1.2 Componenti hardware

- Componenti lato client:
 - Arduino UNO
 - Breadboard
 - Cavi di collegamento
 - Resistori
 - Sensore di prossimità ad ultrasuoni HC-SR04
 - Buzzer
 - Potenziometro
 - Multiplexer CD4067B
 - Button
 - LED verde
 - LED RGB
 - LED rosso 6 pin
 - LED giallo 6 pin
- Componenti lato server:
 - Odroid C2
 - Monitor LCD

1.3 Componenti software

Librerie Arduino:

- NewPing
- ArduinoJson

IDE di sviluppo: Atom con PlatformIO

Linguaggi di sviluppo:

- C++ (Wiring), per lo sketch Arduino lato client

Task:

- SonarTask
- ButtonTask
- BuzzerTask
- LedTask
- LedPwmTask
- LedRgbTask