

# 1 Progetto

L'obiettivo del progetto finale è quello di realizzare un gioco in cui si valutano i riflessi dei giocatori.

Il gioco prevederà che due giocatori impugnino il MicroBit rispettivamente sul tasto A e B. Il microcontrollore fornirà un segnale visivo e che indicherà ai giocatori di premere il tasto, il primo giocatore a premerlo vince.

Il gioco è semplice, ma per implementarlo è necessaria una buona comprensione di come funzionino variabili e cicli.

Per favorire la comprensione del progetto, si consiglia di sviluppare il lavoro in maniera progressiva, facendo realizzare prima una versione a giocatore singolo e poi una a due giocatori (come illustrato nelle sezioni 1.1 e 1.2).

## 1.1 Prima implementazione

L'obiettivo è realizzare un misuratore di tempi di reazione che misuri il tempo che passa tra l'emissione di un segnale dal microcontrollore e la pressione del tasto A da parte del giocatore.

### 1.1.1 [Attività] Identificazione delle 3 fasi principali del programma

Si consiglia di impostare questa attività come discussione con tutta la classe, in modo da favorire la comprensione del progetto e la partecipazione di tutti gli studenti.

Chiedere agli studenti di identificare le tre fasi principali del programma e descrivere cosa deve fare ognuna di queste.

Le tre fasi sono:

#### 1. Attesa / Preparazione

- Mostrare un segnale visivo di attesa (ad esempio una croce);
- Attendere un intervallo randomico di tempo;

#### 2. Misurazione

- Mostrare un segnale visivo di conferma (ad esempio una spunta o un cerchio);
- Iniziare a misurare il tempo;
- Attendere la pressione del tasto A;

#### 3. Visualizzazione

- Calcolare il tempo impiegato dal giocatore;
- Mostrarlo sul display;

### 1.1.2 [Attività] Implementazione della prima fase

La prima implementazione deve funzionare come segue:

1. Il giocatore impugna il MicroBit sul tasto A;
2. Il display del MicroBit mostra un segnale visivo di attesa (ad esempio una croce);
3. Dopo un numero variabile di secondi (in un intervallo scelto dallo studente) il display mostra un segnale visivo di conferma (ad esempio una spunta o un cerchio) ed avvia un timer;
4. Nel momento in cui il giocatore vede il segnale visivo di conferma, deve premere il tasto A; appena viene rilevata la pressione del tasto, il microcontrollore deve fermare il timer e mostrare il tempo impiegato dal giocatore in millisecondi.
5. EXTRA: Impedire che il giocatore possa barare premendo il tasto prima che venga mostrato il segnale visivo di conferma.

In questa prima implementazione, il giocatore gioca da solo e non è prevista una competizione, l'obiettivo è ottenere un punteggio più basso possibile.

È stato inserito un obiettivo extra per permettere agli studenti più veloci di approfondire l'argomento e fornire più tempo agli studenti che possono averne bisogno.

**Soluzione** La soluzione può essere trovata nei file *microbit-3-reaction\_game-1p.hex* e *microbit-3-reaction\_game-1p-extra.hex*

## 1.2 Implementazione finale

Questa implementazione sarà una modifica della prima, in cui verrà aggiunto un secondo giocatore.

L'obiettivo è realizzare un gioco in cui due giocatori si sfidano per vedere chi ha i riflessi migliori.

### 1.2.1 [Attività] Identificazione delle variazioni rispetto alla prima implementazione

Le differenze si trovano in fase di misurazione e visualizzazione.

#### 1. Misurazione

- Attendere la pressione del tasto A o del tasto B;

#### 2. Visualizzazione

- Calcolare quale giocatore ha premuto il tasto per primo;
- Visualizzare A o B sul display per indicare il vincitore;

### 1.2.2 [Attività] Implementazione della seconda fase

La seconda fase deve estendere la prima, aggiungendo la possibilità di giocare in due utilizzando tasto A e tasto B.

La visualizzazione del tempo impiegato verrà sostituita con la visualizzazione del giocatore che ha premuto per primo il tasto.

L'implementazione deve funzionare come segue:

1. I giocatori impugnano il MicroBit sul tasto A e B;
2. Il display del MicroBit mostra un segnale visivo di attesa (ad esempio una croce);
3. Dopo un numero variabile di secondi (in un intervallo scelto dallo studente) il display mostra un segnale visivo di conferma (ad esempio una spunta o un cerchio) ed avvia un timer;
4. Nel momento in cui i giocatori vedono il segnale visivo di conferma, devono premere il tasto; appena viene rilevata la pressione del tasto, il microcontrollore salva il tempo di stop per quel tasto;
5. Il microcontrollore confronta i due tempi e mostra sul display il giocatore che ha premuto per primo il tasto.
6. EXTRA: Impedire che il giocatore possa barare premendo il tasto prima che venga mostrato il segnale visivo di conferma.
7. EXTRA: Mostrare anche lo scarto tra i tempi dei due giocatori (differenza tra i due).

**[Attività] Discussione su possibili soluzioni alternative** Il paragrafo *Soluzione* presenterà due soluzioni, la soluzione A è quella attesa dato l'esercizio, la soluzione B è una soluzione alternativa che non usa il concetto di tempo.

Una volta ottenuta in aula la soluzione A, chiedere se gli studenti riescono a pensare ad una soluzione alternativa che non usi il concetto di tempo.

Valutare vantaggi delle due soluzioni:

- **A**
  - Facilmente espandibile per mostrare anche il tempo impiegato (o lo scarto tra i due tempi);
- **B**
  - Minor numero di variabili (numero di variabili non dipendente dal numero di giocatori);
  - Codice di più semplice comprensione;

**Soluzione** Sono state identificate due soluzioni possibili, salvate nei file *microbit-4-reaction\_game-2p-a.hex* e *microbit-4-reaction\_game-2p-b.hex*.

La prima soluzione è quella attesa, ovvero un'evoluzione della prima implementazione in cui si misurano i tempi di reazione di entrambi i giocatori e si sceglie il giocatore con il tempo minore.

La seconda soluzione, invece, rimuove completamente il concetto di tempo, tenendo traccia direttamente di quale pulsante è stato premuto per primo.

Per quanto non sia un'evoluzione della prima implementazione, è comunque valida e può essere utilizzata per mostrare agli studenti come esistano più soluzioni per un problema.