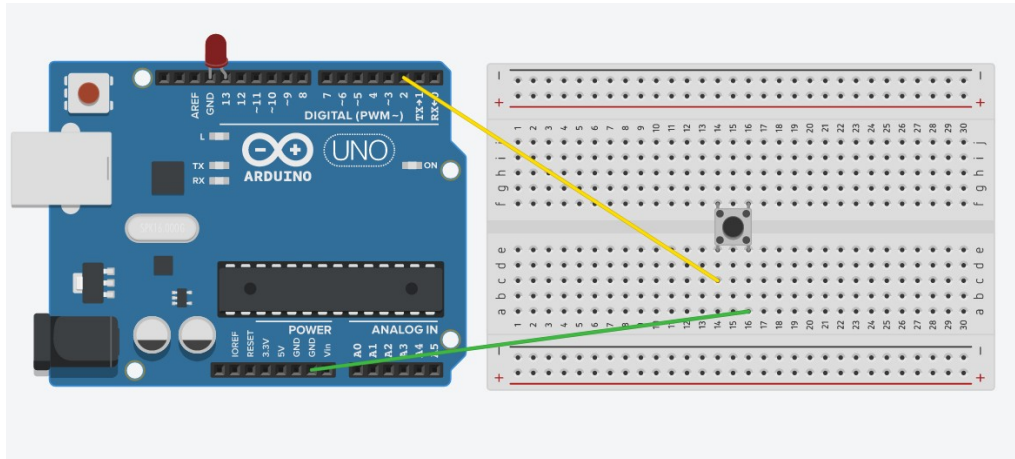


## GIOCO DEI RIFLESSI CON ARDUINO

**Descrizione:** il gioco dei riflessi nella sua forma base consiste in un game in cui l'utente preme un bottone ogni qualvolta il led va in on. Se riesce nell'intento accumula un punto, se fallisce perde una vita. Se fallisce la pressione del tasto per tre volte (ovvero, se l'utente perde le 3 vite a disposizione), il gioco passa in GAME OVER.

**Circuito:** di seguito lo schema circuitale in tinkercad (<https://www.tinkercad.com/circuits>):



**Codice:** [https://github.com/umbertochimenti/makersproject/tree/master/arduino\\_reflexes\\_game](https://github.com/umbertochimenti/makersproject/tree/master/arduino_reflexes_game)

Il codice dopo aver settato come INPUT\_PULLUP il pin 2 (per consentire l'utilizzo del button senza resistenza), setta il pin 13 come OUTPUT (per connettervi il led). Vengono inizializzate le seguenti variabili:

```
//tempi minimi e massimi di accensione/spegnimento del led  
int min = 500;  
int max = 600;
```

```
//punteggio/vite disponibili/variabile di uscita dal gioco  
int points = 0;  
int life = 3;  
bool exit_game = false;
```

All'interno di void loop(), la parte più importante del codice riguarda la gestione del riconoscimento della pressione del tasto durante l'accensione del led:

```
int timeOut = random(min, max);  
bool button_reflex_ok = false;  
digitalWrite(13, 1);  
while(!exit_loop) {  
    if (digitalRead(2) == 0) {
```

```
    button_reflex_ok = true;
}
if(count >= timeOut) {
    exit_loop = true;
}
delay(1);
count++;
}
```

Utilizzando la funzione random(min, max) è possibile generare un numero casuale compreso tra min e max. dopo aver acceso il led (con digitalWrite(13, 1)), se la lettura del bottone premuto va a buon fine (if (digitalRead(2) == 0)) è possibile settare la variabile button\_reflex\_ok a true. Il ciclo while è controllato dalla variabile exit\_loop, che viene settata a true solo allo scadere del timeOut, controllato dall'incremento della variabile count. La temporizzazione del gioco avviene con l'istruzione delay(1) (attesa di 1 ms, prima di ripartire con la verifica della condizione di loop. Un codice molto simile viene utilizzato per gestione il riconoscimento della pressione del tasto durante lo spegnimento del led. Inoltre, se l'utente preme il tasto quando il led è spento ha commesso un errore e verrà penalizzato perdendo una vita (fino al GAME OVER):

```
if(button_reflex_err) {
    life--;
    Serial.println("PRESS ERROR!!!!");
    if(life == 0) {
        exit_game = true;
        Serial.println("GAME OVER!!!");
    }
}
```