



**Reviving hands-on educational play for  
learning skills of tomorrow**  
PROJECT N° 2019-1-UK01-KA201-061466

**MODULO 2**

**Scratch 2.0 – DIY Kit Elettronico**

**SVILUPPATO DA IDEC & PLATON**

**CIVIC**

 **idec**

  
ΕΚΠΑΙΔΕΥΤΗΡΙΑ  
ΠΛΑΤΩΝ

 **learn**  
EUROPEAN DIGITAL LEARNING NETWORK

  
**scholé**

**Emphasys**  
CENTRE

 **CCS**  
Digital Education

### DESCRIZIONE DEL MODULO

Scratch è uno strumento di programmazione visiva. Utilizzando Scratch è possibile creare animazioni e giochi con un'interfaccia *drag-and-drop*. In questo modo si potranno creare senza difficoltà giochi per computer e storie interattive senza aver bisogno di scrivere il codice.

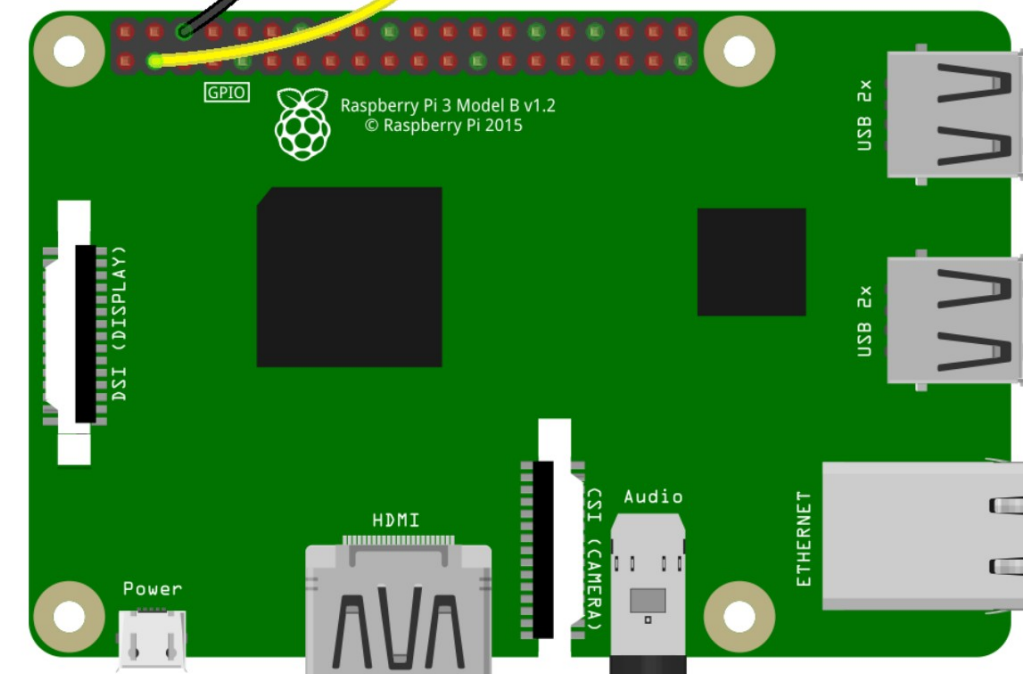
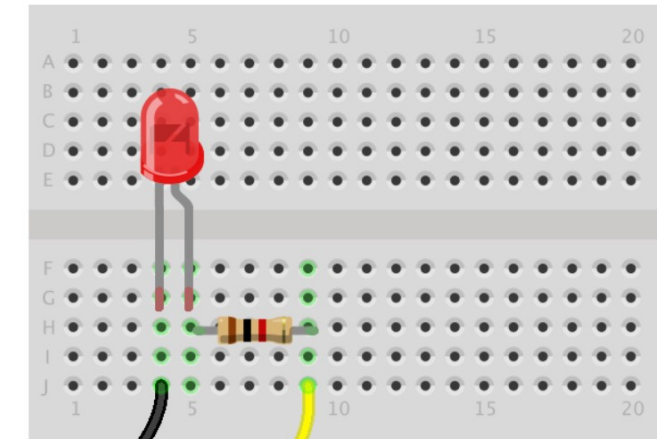
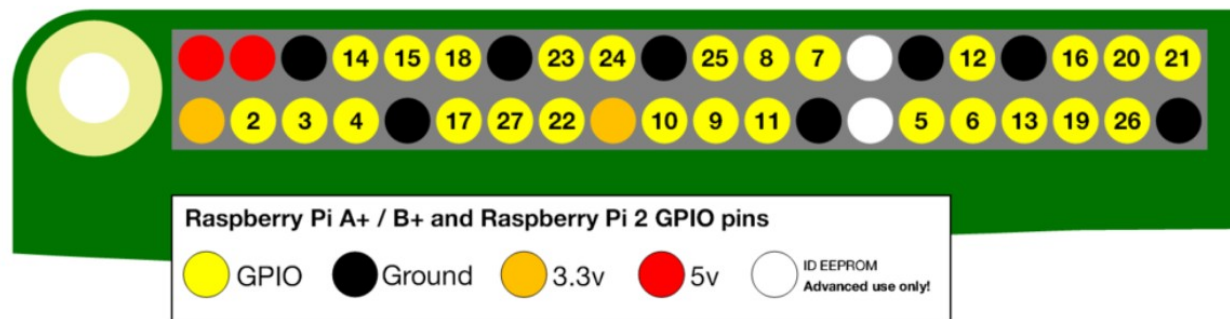
La versione di Scratch disponibile con il Raspberry Pi rende semplice la comunicazione tra il Raspberry e i pin GPIO (General Purpose Input Output). Attraverso questi pin è possibile connettere il proprio Raspberry Pi a sensori, luci led, pulsanti e molto altro.

Cercheremo di presentarvi una breve dimostrazione utilizzando il Pi e i suoi componenti. Per farlo adopereremo Scratch per programmare il Pi con l'obiettivo di far lampeggiare un LED.

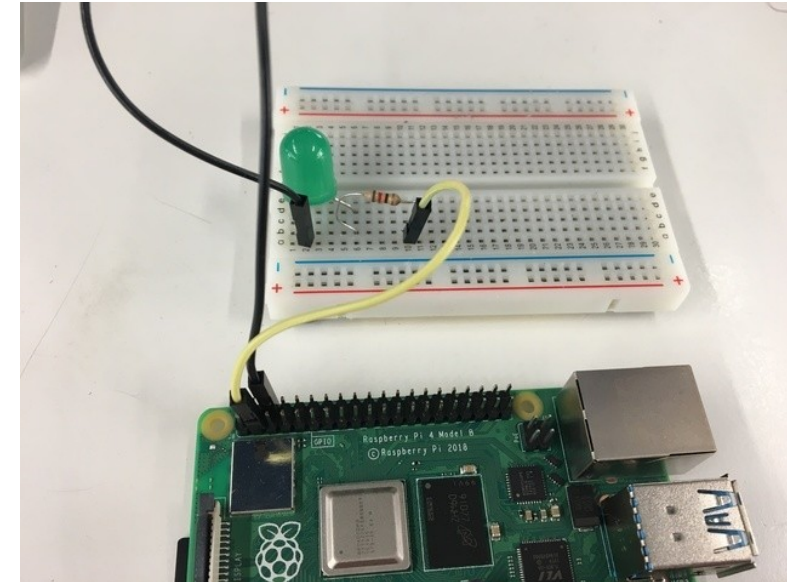
**Prima di tutto, creiamo i collegamenti**

Dovrete avere a portata di mano:

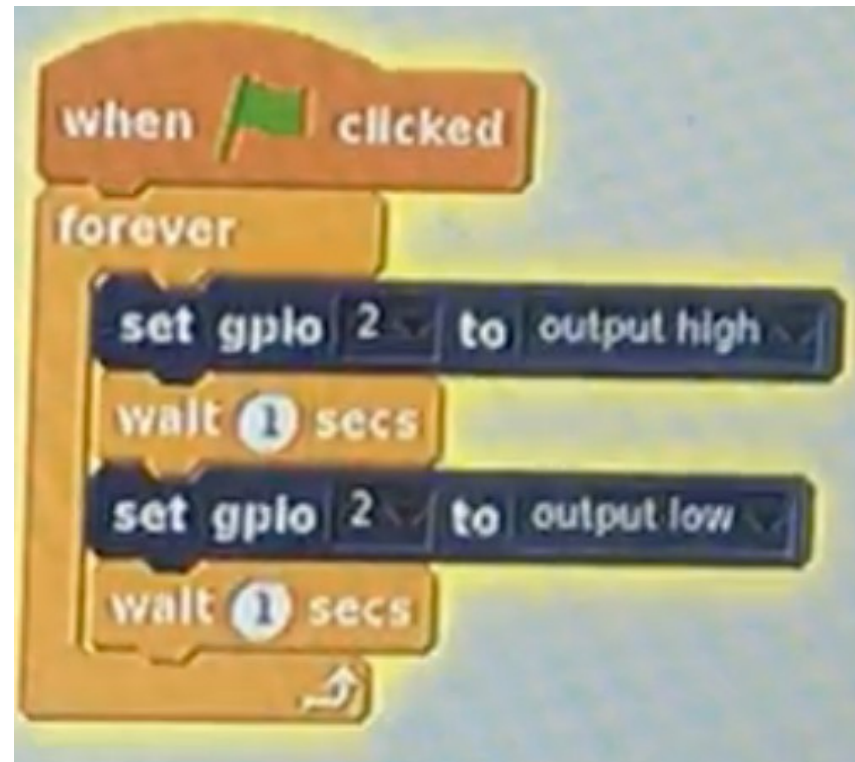
- Un led (di qualsiasi colore)
- Un resistore uguale o inferiore a 1K
- 2 cavetti jumper, maschio e femmina



- Collegare un cavetto jumper femmina a uno maschio dal pin 2 GPIO sul Pi su qualsiasi punto della breadboard.
- Poi, collegare il resistore (1K o meno è sufficiente) dal cavetto jumper GPIO 2 a un altro punto della breadboard
- Ora, esaminate il LED e determinate qual è la parte più lunga, questa andrà collegata all'altra parte del resistore con l'estremità opposta che andrà inserita in un'altro punto della breadboard.
- Infine, collegare i cavetti jumper maschi e femmine dall'estremità non connessa del LED al pin del ground sul Pi.



Poi, programmiamo il Pi in scratch per far lampeggiare il LED.



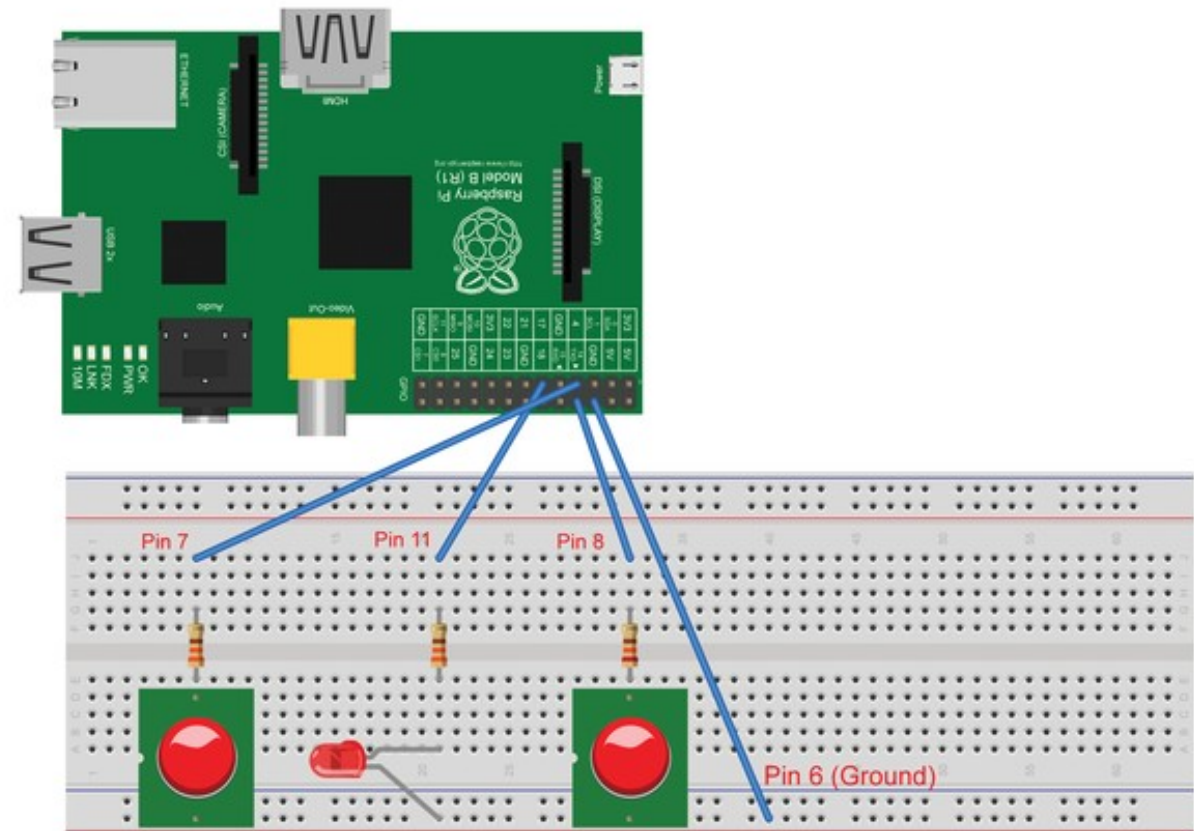
Il tuo LED sta **LAMPEGGIANDO**

Ora cercheremo di presentarvi una breve dimostrazione utilizzando il Pi e i suoi componenti. Per farlo adopereremo Scratch per programmare il Pi per creare un gioco multigiocatore di reazione rapida agli eventi.

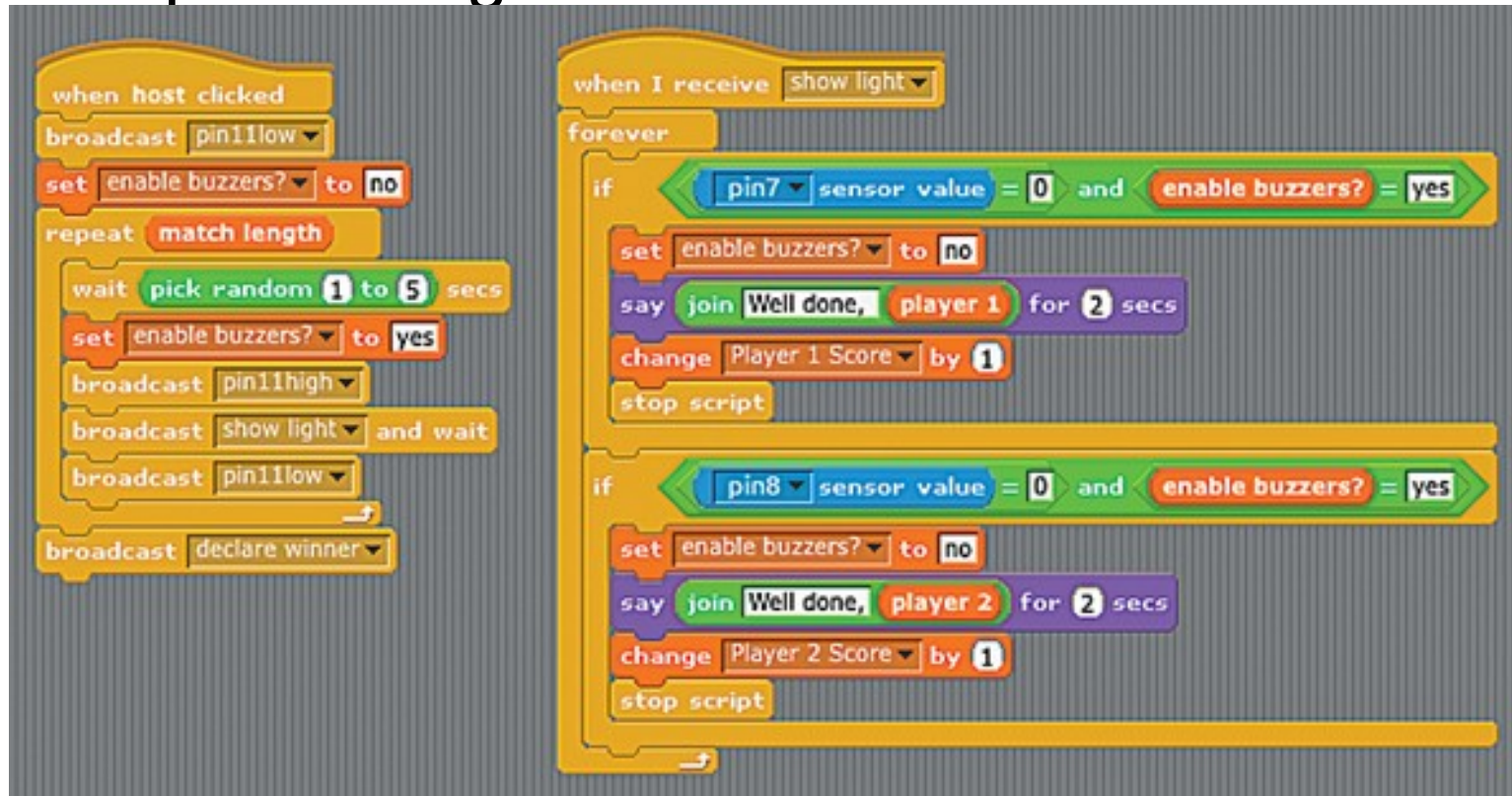
Per vedere quale giocatore ha una reazione più rapida, è necessario collegare due buzzer alla breadboard. Il programma Scratch controllerà quando i giocatori sono autorizzati a premere il pulsante, rilevando la prima pressione sul buzzer, e tenendo il punteggio.



Per il buzzer si useranno due pulsanti, uno per ciascuno dei due giocatori. Il buzzer del primo giocatore si collegherà al pin 7, e quello del secondo al pin 8. (Guarda la figura sulla destra per i circuiti del buzzer). Per rilevare quando il buzzer viene premuto, lo script dello Scratch ha bisogno di monitorare l'input nei pin del Raspberry Pi. Lo Scratch GPIO considera come input i pin 3, 5, 7, 8, 10, 19, 21, 22, 23, 24, e 26.



I due script mostrano la principale logica del gioco. Lo script inizia con **quando l'host ha cliccato sul** blocco avvia il gioco. Ho aggiunto una nuova variabile chiamata **abilitare i buzzer?** a cui ho assegnato il valore **no**, per impedire ai giocatori di trasmettere la vibrazione troppo presto.



```
when host clicked
  broadcast pin11low
  set enable buzzers? to no
  repeat match length
    wait pick random 1 to 5 secs
    set enable buzzers? to yes
    broadcast pin11high
    broadcast show light and wait
    broadcast pin11low
  broadcast declare winner

when I receive show light
  forever
    if pin7 sensor value = 0 and enable buzzers? = yes
      set enable buzzers? to no
      say join Well done, player 1 for 2 secs
      change Player 1 Score by 1
      stop script
    if pin8 sensor value = 0 and enable buzzers? = yes
      set enable buzzers? to no
      say join Well done, player 2 for 2 secs
      change Player 2 Score by 1
      stop script
```

Il loop **ripete** () usa una variabile chiamata **match length** per controllare quante partite o set avrà ogni gioco. Questo approccio permette ai giocatori di raggiungere almeno le migliori due su tre, per esempio. Questa configurazione può essere controllata da un comando di controllo sul palcoscenico.

Dopo un casuale tempo di attesa compreso tra **1** e **5** secondi, lo script abilita i suoni e poi accende il LED trasmettendo un messaggio **pin11high**. Quando il LED si accende, questo è il segnale per i giocatori di fare rumore. La **trasmissione (show light)** e il **blocco di attesa** coordinano la programmazione necessaria per rilevare i buzzer dei giocatori.

Lo script **quando ricevo (show light)**, rileva l'input del pin. Il loop **sempre** assicura che lo script continuerà a funzionare finché il giocatore uno o il giocatore due cliccheranno sul buzzer. Il valore del blocco del **() sensore** ha delle opzioni a tendina per gli input di ScratchGPIO.

Inoltre, il primo blocco **se()** controlla se il pin 7 (giocatore 1) è acceso e se la variabile **abilitare il buzzer?** è uguale a **sì**. Se entrambe le condizioni sono vere, lo script incrementa il punteggio del giocatore uno, disabilita il buzzer, e ferma lo script corrente. Il secondo blocco **se()** esegue lo stesso controllo per il pin 8 (giocatore 2).

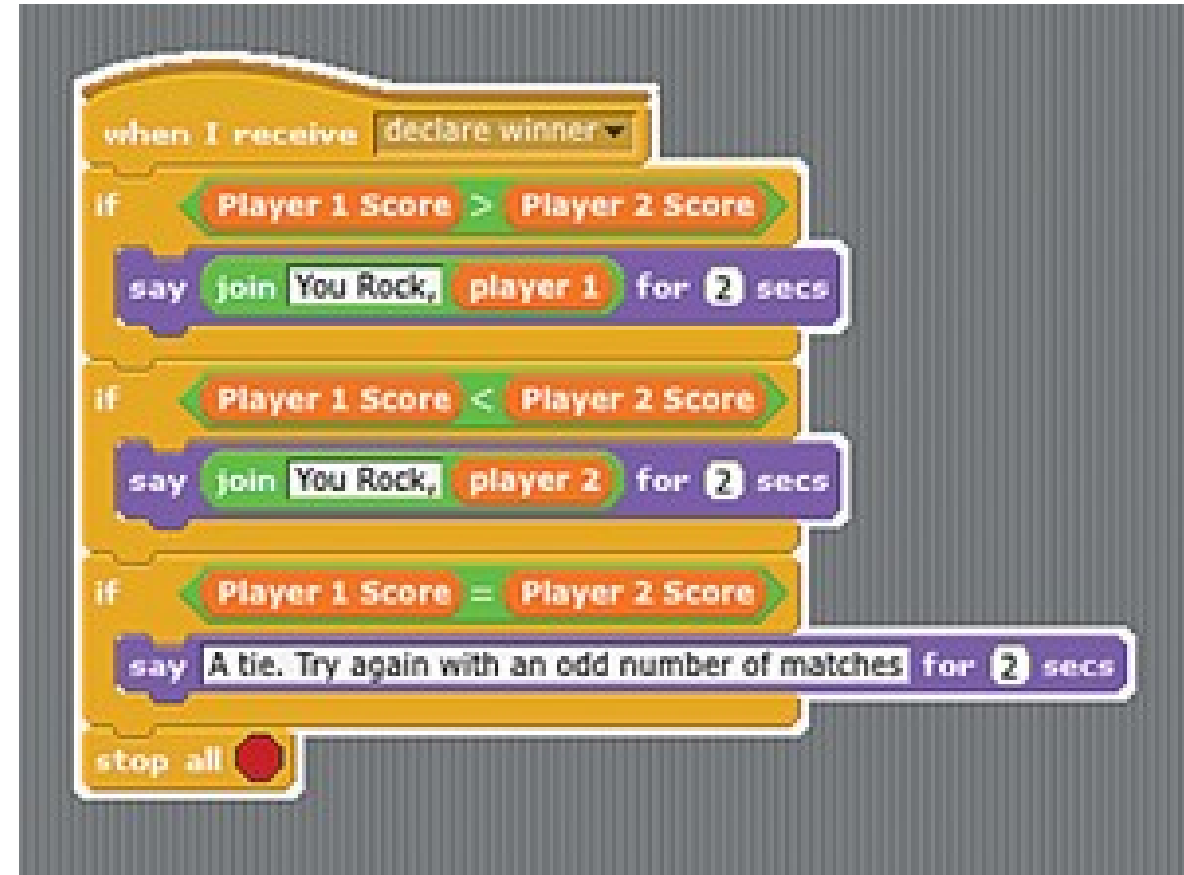
I pin di input hanno un valore predefinito di **1** nella posizione "off". Quando l'input viene rilevato, per esempio cliccando sul buzzer, il valore del pin diventa **0**. Lo si può vedere abilitando il valore del sensore **pin7** nel monitor del palcoscenico.

Disabilitando il buzzer si impedisce agli altri giocatori di fare rumore e ottenere un punto. Fermando lo script si spezza il loop **sempre** e si segnala **quando l'ospite ha cliccato** lo script per ripartire da dove ci si era fermati, cosa che consisterà nell'eseguire il blocco **broadcast(pin11low)** per spegnere il LED. Il vantaggio di usare il blocco **broadcast(show light) and wait** è che lo script sarà in pausa finché tutti gli script **when I receive(show light)** saranno completati.



Poi, lo script continua a illuminare casualmente il LED per un numero stabilito di partite. Alla fine del gioco, il programma dichiara il vincitore con il blocco **broadcast(dichiara vincitore)** alla fine di quando l'ospite ha cliccato lo script.

Lo script **quando io ricevo(dichiara vincitore)** determina che giocatore ha il punteggio più alto e annuncia un vincitore.



# GIOCHIAMO!

Attenzione a non rompere i pulsanti...

## Bibliografia

<https://learn.adafruit.com/programming-with-scratch-on-raspberry-pi/demo>

[https://www.raspberry-pi-geek.com/Archive/2014/07/Creating-a-multiplayer-quick-reaction-game/\(offset\)/2](https://www.raspberry-pi-geek.com/Archive/2014/07/Creating-a-multiplayer-quick-reaction-game/(offset)/2)



<https://www.instructables.com/id/Physical-Computing-Scratch-20-for-Raspberry-Pi/>



**Reviving hands-on educational play for  
learning skills of tomorrow**  
PROJECT N° 2019-1-UK01-KA201-061466

