



# PLINKO!

---

Progetto Raspberry Pi

**Artifoni Matteo,  
Shkina Fabian,  
Paletta Andrea,  
Inzoli Leonardo**

# Scopo

---

Progettare un gioco fisico interattivo che unisce **Arduino** per la parte hardware e **Raspberry Pi** per l'interfaccia e il controllo.

PLINKO!



# Motivazioni e contesto

Abbiamo scelto di realizzare PLINKO! perché volevamo:

- Unire **programmazione** e **progettazione fisica** in un unico progetto concreto e interattivo.
- Sperimentare la collaborazione tra **Arduino** (**controllo fisico**) e **Raspberry** (**gestione interfaccia**).
- Introdurre elementi di **gioco e scommesse simulate**, per rendere il progetto più coinvolgente e realistico.



# Obiettivi

---

- Costruire una **struttura fisica** di PLINKO funzionante.
- Integrare Arduino per gestire:
  - **Il rilascio della pallina** (**servomotori**).
  - **La rilevazione del punto di arrivo** (**sensori**).
- Creare un **interfaccia via web** accessibile da PC e telefono con Raspberry Pi.
- Permettere all'utente di scommettere e calcolare la vincita.





# Applicazioni

- Progetto didattico interdisciplinare (informatica, telecomunicazioni).
- Esempio pratico di **Internet of Things**: un gioco fisico controllabile via rete.
- Potenziale utilizzo in fiere scolastiche o open day per dimostrare l'interazione uomo-macchina.



# Tecnologic

e



- La precisione dei sensori può influenzare la rilevazione del punto di arrivo.
- I servomotori richiedono un'alimentazione stabile per un corretto funzionamento.
- La comunicazione tra Arduino e Raspberry deve essere sincronizzata correttamente e stabile.
- La pallina non risale da sola per poter essere rilanciata



# Funzionalità

- Avvio del gioco da **interfaccia web** (via Raspberry).
- Arduino attiva i servomotori per il rilascio della pallina.
- I sensori leggono il punto di arrivo e inviano il risultato.
- Confronto con la scommessa effettuata.
- Interfaccia utente mostra esito e **vincita**.



# CODICE

Ora vediamo la parte pratica del codice! Quello del Raspberry Pi è diviso in moduli separati, ognuno con uno scopo preciso. Questo migliora la leggibilità, l'organizzazione e il controllo degli errori.

Inoltre per permettere concorrenza e un sincronismo più preciso, abbiamo scelto di utilizzare i thread per l'esecuzione delle varie funzionalità, nello specifico:

- Un thread per la connessione bluetooth con l'ESP
  - Un thread per le pagine tramite Flask (che tecnicamente è già un thread a sé stante)
  - Un thread per seguire l'andamento della partita iniziata, che contiene le informazioni del giocatore (username e puntata), che termina insieme alla partita dopo la comunicazione della vincita





# CODICE

Prima di poter accedere alla pagina principale, viene gestita la parte di login/registrazione.

Abbiamo pensato fosse interessante implementarla così che ogni giocatore possa avere il proprio credito e informazioni salvati, e creare un contesto più reale.

I dati vengono salvati in un file json sul raspberry, che simula un database.



# CODICE

Dopo aver fatto l'accesso, la pagina principale ci indica lo stato di connessione bluetooth (che avviene automaticamente nel thread parallelo separato), costantemente aggiornata tramite richieste AJAX, a una delle route flask.

L'utente può ricaricare il proprio conto (con un'interfaccia che emula l'inserimento dei dati di una carta di credito), e, se la connessione, è presente può scegliere di avviare la partita.



cosa è una richiesta ajax

Una **richiesta AJAX** (Asynchronous JavaScript And XML) è un modo per fare chiamate al server in modo asincrono, cioè **senza ricaricare l'intera pagina web**.

# CODICE

Quando l'utente avvia la partita, viene creato un thread apposito che ne segue l'andamento, anche nel caso dovesse chiudere la pagina.

Viene inviato all'ESP il messaggio '1' per segnalare l'inizio della partita. Questo aziona il servomotore per far cadere la pallina, la cui caduta può avvenire in 7 caselle diverse, identificate con messaggio da '2' a '8' che viene mandato una volta sola al raspberry.

Quando viene ricevuto, la variabile globale che segnala se si è in partita viene settata a false e in base a cosa si riceve si calcola quanto il giocatore ha vinto o perso. Questo viene salvato nei dati del giocatore nella sezione 'events', che viene letta dalla pagina flask quando si ha la pagina aperta (con la route /checkEvento) e segnala l'esito della partita, anche nel caso si riapra la pagina dopo averla chiusa.



### Accedi

  
  
☐ Rimani loggato  

Non hai un account? [Registrati qui](#)

Login

```
{
  "users": {
    "matteo": {
      "password": "*****",
      "name": "Matteo",
      "surname": "Artifoni",
      "credit": "1070.0",
      "event": ""
    },
    "ciro": {
      "password": "*****",
      "name": "ciro",
      "surname": "esposito",
      "credit": "12781.0",
      "event": ""
    }
  }
}
```

✖ Errore nella connessione Bluetooth

Ciao matteo! Credito: €1070.0 Logout

### Benvenuto al Gioco

  

Avvia Gioco

Ricarica con carta

### Inserisci dati carta di credito

  
  
  
  

Paga



# Obiettivi Raggiunti

- ✓ Progettazione struttura fisica del gioco.
- ✓ Collegamento e test dei servomotori con Arduino.
- ✓ Rilevazione punto di arrivo con sensori.
- ✓ Creazione interfaccia web su Raspberry.
- ✓ Comunicazione tra Arduino e Raspberry Pi.
- ✓ Emulazione di un contesto reale per gioco/scommesse



# Sviluppi Futuri

Miglioramento dell'estetica del gioco e della struttura.

Aggiunta di effetti sonori o LED in base all'esito.

Implementazione di un sistema di punteggio o classifica.

Riconoscimento vocale per avviare il gioco.

Controllo e feedback utente per gli errori più rigoroso.

Possibilità di scegliere se giocare 'Live' (con la struttura reale), o 'Virtual', simulando la pallina e l'ambiente tramite codice e non realmente.

Pallina che torna da sola in posizione e telecamera per vedere andamento.



# Fonti Utilizzate

- Arduino Documentation
- Raspberry Pi Docs
- YouTube tutorial su progetti PLINKO, Flask e bluetooth
- Stack Overflow (per riscontro con errori di altri analoghi)
- ChatGPT (per supporto tecnico e idee)
- Supporto del prof (!!!)



# Conclusione

---

PLINKO! è stato un progetto che ci ha permesso di **imparare** concretamente:

- A lavorare in **team**.
- A integrare **hardware** e **software**.
- A risolvere problemi reali e non simulati.  
È stato sia una sfida che un divertimento... proprio come il gioco stesso!

