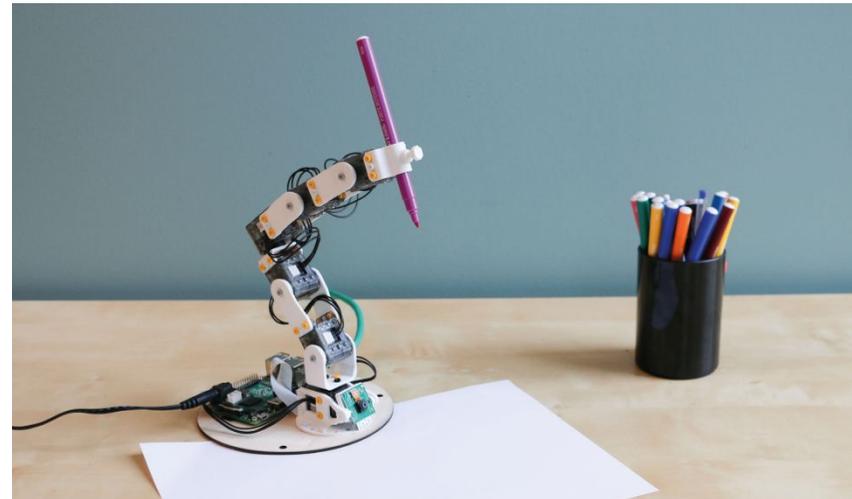


EARLY Teaching Scenario

Tema/Argomento: disegnare con Poppy Ergo Jr

Obiettivi: Gli studenti impareranno a:

- Assemblare il robot Poppy Ergo Jr
- Usare i comandi base di Snap
- Programmare movimenti semplici di Poppy



Abilità che gli studenti svilupperanno durante lo svolgimento dello scenario: connesso con l'indirizzo di scuola →

Il National Curriculum polacco per la scuola primaria stabilisce che entro la fine della terza media, gli studenti dovranno essere in grado di:

- Progettare, creare e testare programmi nel processo di risoluzione di problemi;
- Usare nella programmazione: istruzioni di input / output, aritmetica ed connettivi logici, istruzioni condizionali, funzioni;

- Progettare, creare e testare software che controllano un robot o un altro oggetto sullo schermo o in una situazione reale;
- Cercare in rete informazioni necessarie a svolgere un compito, usare forme complesse di domande e avanzate capacità di ricerca dei motori di ricerca.

Le componenti che sono potenziate/sviluppate in questo teaching scenario sono le seguenti:

- Creare un robot con le componenti fornite;
- Programmare il robot con linguaggio di programmazione a blocchi grafici;
- Problem solving;
- Collaborazione.

Target del gruppo: studenti della scuola secondaria di primo grado.

Età degli studenti: 13 - 14 anni.

Numero di studenti: massimo 10 per sottogruppo.

Durata (durata prevista/numero di lezioni): 3 sessioni di 45 - 90 minuti ciascuna.

Prerequisiti (materiali necessari e risorse online):

- Poppy Ergo Jr con portamatita;
- Computer con parametri sufficienti per eseguire Snap!
- Materiali di supporto online disponibili su <https://www.poppy-project.org/en/robots/poppy-ergo-jr> .

Introduzione dello scenario (*incl. possibili applicazioni, alternative, rischi*)

Poppy Ergo Jr è un braccio robotico pensato per scopi didattici ed è composto da sei motori che permettono un range di movimenti ed elementi di stampa 3D. È dotato di tre strumenti che permettono differenti interazioni con l'ambiente: una pinza, un paralume, un portamatite. Il robot è controllato da un Raspberry Pi board e una videocamera gli permette di interagire con gli oggetti circostanti. I suoi movimenti possono essere programmati con uno strumento di programmazione visuale che permette agli studenti una varietà di esperienze d'apprendimento creative e ludiche. Tuttavia i produttori sostengono che il robot sia facile da

costruire/sviluppare/modificare ed ogni cosa in lui è trasparente e accessibile per promuovere un approccio educativo abbiamo scoperto che Poppy Ergo Jr rappresenta un numero di sfide per una esperienza di workshop di successo. Il seguente scenario propone un percorso/ tracciato da seguire per evitare gli ostacoli sfruttando il potenziale di un braccio robotico nella scuola secondaria.

Prima della programmazione (lavoro preparatorio per gli insegnanti)

Preparare un laboratorio informatico con sufficienti postazioni di lavoro per gli studenti.

- Poppy si basa su Snap! (<https://snap.berkeley.edu/>), una variante di Scratch. Assicurarsi che tutti i computer abbiano una connessione Internet stabile e su di essi possa essere eseguito (aprire la finestra di un browser e connettersi a <http://snap.berkeley.edu/run>).
- Il numero dei bracci robotici disponibili determinerà la grandezza dei gruppi da coinvolgere nel workshop. Si consiglia di far lavorare all'assemblaggio del robot massimo 2-3 studenti. La maggior parte dei componenti da assemblare potrebbe essere stampata in 3D per ridurre i costi della costruzione del robot – stampando solamente il numero necessario di questi componenti e acquistando solo i dispositivi elettronici e meccanici richiesti.
- Prima di tutto, costruisci e programma il robot da solo – consigliamo di dedicare almeno una settimana per questa attività ludica e coinvolgente prima di proporre l'attività agli studenti.

Parte principale dello scenario (3 lezioni)

Ognuna delle lezioni che seguono è una distinta unità di apprendimento che può essere svolta in modo indipendente. Idealmente, tutti gli studenti progrediscono nei vari step ma questo dipende dal loro livello e dal numero dei computer e dei robot disponibili per questa attività. Forse l'idea migliore sarebbe quella di dedicare la prima lezione all'assemblaggio del robot con un piccolo gruppo dedito alla costruzione di 1- 3 robot per poi procedere, nelle lezioni a seguire, con la costruzione del robot da parte dell'intera classe.

Lezione 1: costruire Poppy Ergo Jr

I produttori hanno pubblicato una breve guida che spiega come assemblare il robot, comprendente la configurazione dei motori, l'assemblaggio elettronico e la costruzione dell'hardware

<https://docs.poppy-project.org/en/assembly-guides/ergo-jr/index.html>. questa risorsa sarà il quadro di riferimento principale durante tutto il processo ma non si presta alla lettura integrale da parte degli studenti. Abbiamo sviluppato un video tutorial per principianti in modo da facilitare la costruzione del robot <https://edurobots.eu/assembling-poppy-ergo-jr/>. iniziare guardando il video integralmente. Poi guardarlo una seconda volta e iniziare l'assemblaggio. In caso di incertezze, rivolgersi all'insegnante.



[Motors configuration](#): per connettere i motori e identificarli sulla stessa linea, essi devono avere un unico ID.



Electronic assembly: inserire la micro-SD card nel Raspberry Pi – spingere la micro-SD nello slot connettore fino a quando si sentirà il suono "click".

Mechanical assembly: l'assemblaggio meccanico sfrutta speciali rivetti necessari per connettere le diverse parti del robot. L'orientamento/la direzione di assemblaggio è importante quindi è necessario essere sicuri di aver seguito la guida attentamente.

È una buona idea iniziare lasciando che i ragazzi si cimentino sul robot già montato come fattore motivante. Così saranno più appassionati e avranno la pazienza necessaria per assemblare il robot da soli.



Infine gli studenti testeranno i loro robot andando sulla pagina <http://poppy.local>. Dovranno cliccare sul bottone *Reboot the robot* per essere sicuri che il software del robot sia collegato ai motori. Andando sul monitor, cliccare poi su *Monitor and Control*. Se il software è stato correttamente avviato, dovrebbe essere visibile il logo di connessione in verde, altrimenti sarà rosso. Se ciò accade, bisogna verificare dove è stato fatto un errore guardando nei messaggi della pagina *What happened?*. Nella gran parte dei casi un filo è scollegato o non è stato configurato un motore.

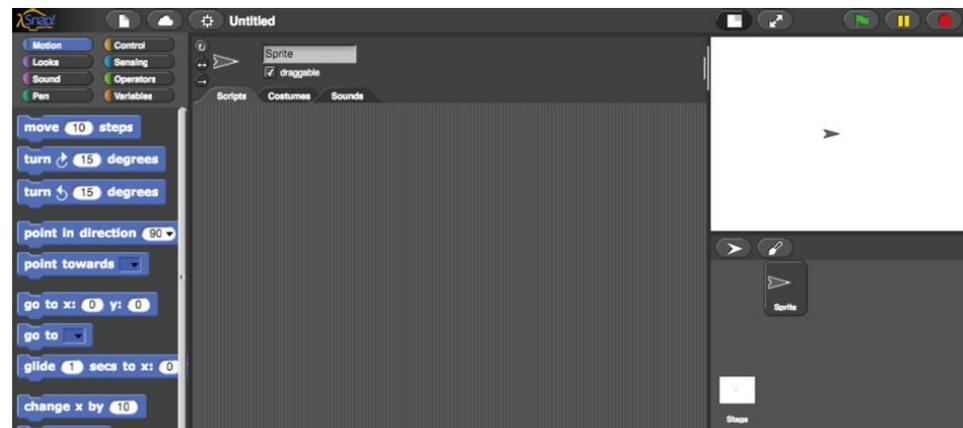
Lezione 2: Connettere Poppy Ergo Jr a Snap!

Snap! È un linguaggio di programmazione grafico a blocchi che permette agli utenti di creare animazioni interattive, giochi e molto altro, mentre si imparano concetti di matematica e di computazione/calcolo. Snap! è open-source ed è interamente scritto in javascript, può essere usato direttamente dal [official website](#) o dalla [copy of the website](#) sul proprio computer, aprendo il file snap.html dal browser. Gli studenti dovranno conoscere le funzioni base di Snap! Per poter programmare il robot Poppy. È questo l'obiettivo di questa lezione.

Per iniziare, gli studenti selezionano Snap! Dal menu principale.

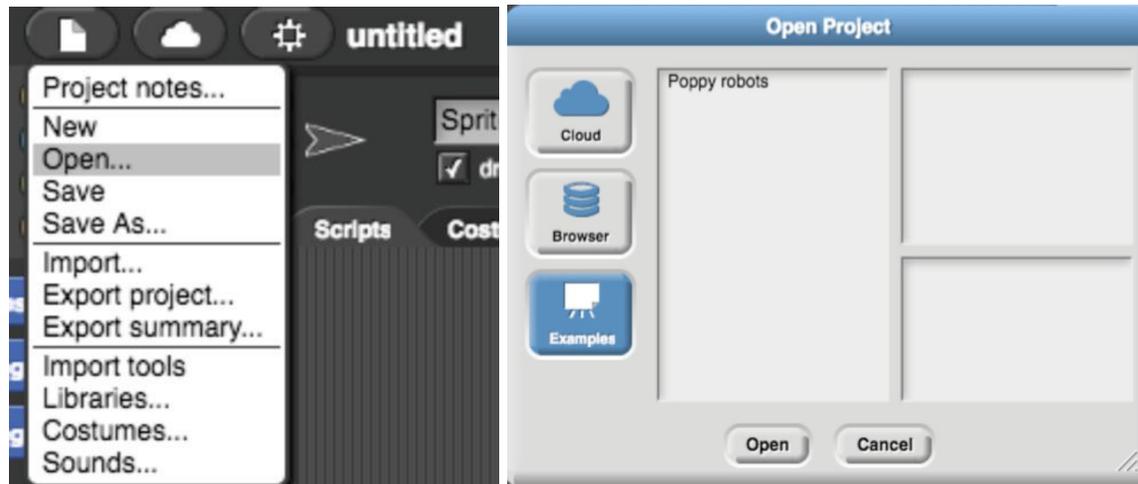
Snap! È diviso in 3 parti: per programmare Poppy Ergo Jr, si usano principalmente le zone a sinistra e al centro.

- A sinistra è presente la lista dei blocchi disponibili ovvero le istruzioni, divise per categorie.
- Al centro vi è l'area dello script in cui vengono assemblati i blocchi.



- In alto a destra, l'area del display è controllata dallo script, sotto vi sono gli oggetti programmabili, che si modificano nell'area di visualizzazione/ display.

Aprire il progetto Poppy robots per poter accedere ai blocchi specifici per Poppy cliccando su: File> Open> Examples> Poppy robots> Open



Suggerimento: per informazioni su Snap! Premere il tasto destro su un blocco e selezionare help.

Nell'area dello script dovrebbero apparire due blocchi.

Verifica se sei connesso al robot:

- Scrivi il nome del robot o l'indirizzo IP nel blocco (qui il nome del robot è poppy);

- Clicca sul blocco TEST CONNECTION per verificare che sei correttamente connesso al robot;
- Se compare un messaggio di errore, verifica di essere connesso correttamente (se necessario, controlla la documentazione del robot).

Salvare il programma: è possibile salvare il programma sull'hard drive del computer cliccando su: Files> Export> right click> Save as

oppure salvarlo nel cloud, con il proprio user account:

- Clicca sul menu del cloud sulla barra degli strumenti;
- Seleziona l'opzione iscriviti/registratori dal menu e segui le istruzioni;
- Verifica nella posta elettronica di aver ricevuto la password iniziale d'accesso.

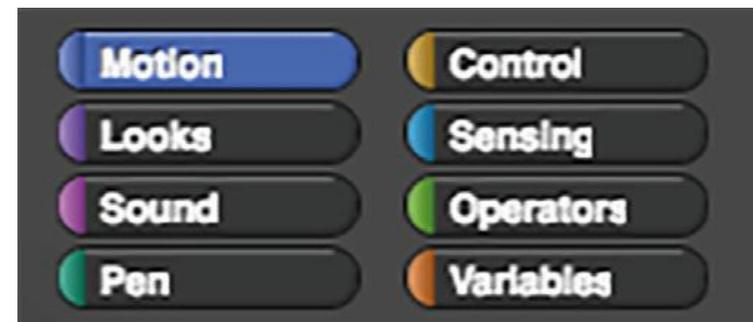
Prima di attivare Ergo Jr, controlla la posizione del robot ogni volta. A questo punto sei pronto per iniziare le attività!

Lezione 3: come disegnare con Poppy Ergo Jr

In questa lezione gli studenti impareranno come far disegnare il robot usando Snap! - un linguaggio di programmazione che assembla istruzioni mediante blocchi. Un assemblaggio di blocchi si chiama script.

Per trovare i blocchi su Snap ! puoi cercarli mediante:

- Colore/ categoria (ad ogni colore corrisponde una categoria);
- Parola chiave: > Right click on the left side> find blocks, che permette anche di selezionare solo i blocchi specifici per Ergo Jr.



Crea i due script qui sotto per far assumere a Ergo Jr posizioni specifiche:

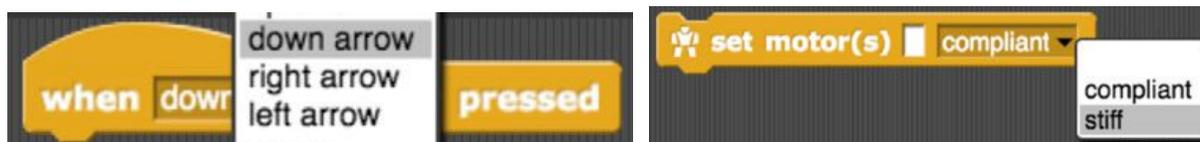


Per farlo:

1. Seleziona e rilascia i tre blocchi di cui hai bisogno nello spazio centrale:



2. Clicca sulle liste e scegli i valori corretti:



3. Assembla i blocchi:



4. Fai la stessa cosa con il secondo script.

Per nidificare i blocchi: seleziona il blocco e rilascialo nel posto corretto con il mouse. Il bordo bianco indica che i blocchi si stanno unendo.



Puoi copiare/incollare i blocchi e gli script: cliccando sul tasto destro, poi su duplicare.



Per attivare entrambi gli script (il bordo dello script attivato diventa bianco): Premi \downarrow o \uparrow sulla tastiera e manovra il robot per fargli assumere posizioni diverse.

Per muovere Ergo Jr motore per motore usiamo il seguente blocco:



Prima di far muovere il robot con Snap! assicurati che tutti i motori siano attivati (stiff mode) e mettili nella posizione iniziale, che corrisponde alla posizione 0 di ogni motore. Per farlo, esegui il seguente blocco:



Posiziona il motore m1 a 90 gradi in 2 secondi:



Esegui i blocchi opposti a loro volta:



I blocchi hanno forme differenti, ogni forma corrisponde a una categoria specifica:

- I blocchi ovali () si chiamano Reporters: quando vengono eseguiti restituiscono un valore.

- All'inizio dello script può esserci un Hat block, che indica quando lo scenario deve essere eseguito. Solitamente iniziano con

la parola When (); uno script non deve necessariamente avere un Hat block, ma, in sua assenza, lo script sarà eseguito solo se l'utente clicca su di esso.

- I blocchi Command () corrispondono a un'azione.

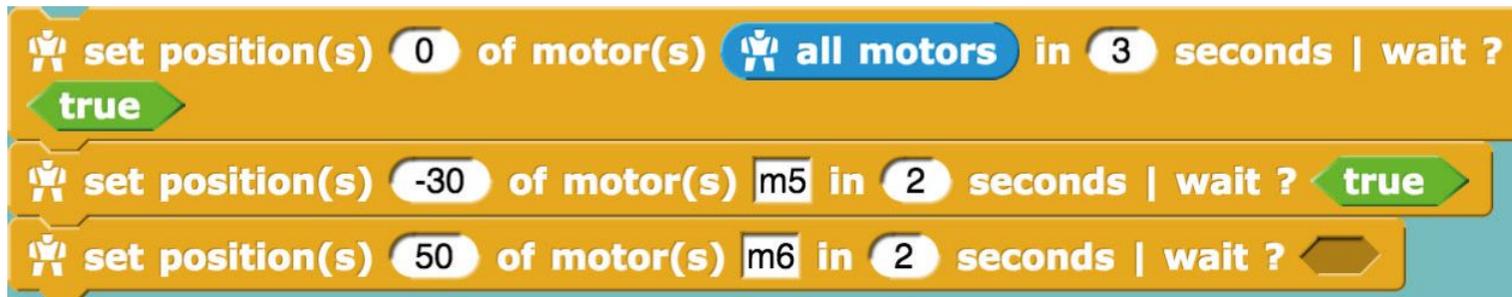
Modifica il blocco Set position per posizionare il motore m1 e il motore m5 nella posizione -30 gradi in 2 secondi.

Con l'aiuto dei blocchi che abbiamo appena scoperto costruisci due programmi che corrispondono alle seguenti istruzioni:

- Premi ⇨ sulla tastiera, poi metti tutti i motori nella posizione di 0 gradi in 3 secondi;
- Premi ⇨ sulla tastiera, poi metti il motore m1 e il motore m4 in posizione 60 gradi in 2 secondi.
-

Ora usiamo i motori per creare i movimenti.

Esegui lo script sottostante e guarda cosa succede:



The image shows a Scratch script with three 'set position' blocks. The first block is: 'set position(s) 0 of motor(s) all motors in 3 seconds | wait ? true'. The second block is: 'set position(s) -30 of motor(s) m5 in 2 seconds | wait ? true'. The third block is: 'set position(s) 50 of motor(s) m6 in 2 seconds | wait ?'. The 'wait ?' field in the third block is currently empty.

Sostituisci il secondo blocco  con il blocco 



Quando ci sono due o più blocchi nidificati, in che ordine avvengono le azioni? Cosa succede quando Wait è settato su true? Cosa succede quando Wait è settato su false?

Le linee del codice vengono eseguite pressoché istantaneamente e, qualche volta, anche se la linea precedente non è ancora stata raggiunta. Wait permette di aspettare fino a quando il motore non ha raggiunto la posizione richiesta eseguendo il comando successivo.

Con i blocchi che ora conosci, programma Ergo Jr in modo che compia un movimento di saluto quando premi la b key.

- Inizia con un movimento semplice e arricchiscilo strada facendo.
- Scegli i motori che vuoi usare per creare il movimento.
- Fai in modo che il robot compia il movimento e osserva le azioni di ogni motore.
- Puoi aiutarti con il blocco  per sapere la posizione del motore e notarne il valore per poterlo riutilizzare.
- Programmi il movimento un motore per volta e testa ogni volta il risultato del programma.

Non esitare a creare e testare altri movimenti!

Risultati di apprendimento

Gli studenti saranno capaci di:

- Lavorare autonomamente e credere in se stessi;
- Seguire istruzioni scritte e video tutorials;
- Assemblare in piccolo braccio robotico con componenti forniti;
- Usare un linguaggio di programmazione a blocchi per far muovere il robot;
- Collaborare con i compagni nelle fasi dell'attività.

Fonti

Questo scenario è basato sui materiali reperibili su <https://www.poppy-project.org/en/robots/poppy-ergo-jr>:

- Apprendre à programmer Ergo Jr en Snap! Livret d'accompagnement du robot Poppy Ergo Jr (Conception et réalisation : Équipe Flowers Inria et Ensta ParisTech et Poppy Project; Design graphique : Antonin+Margaux)
- A day to build and program your robot! (<https://www.poppy-project.org/en/posts/poppy-ergo-jr-workshop-at-cern>)