

Robotica educativa ed intelligenza artificiale *Thymio e Scratch*

SCHEDE ATTIVITÀ DIDATTICHE PER INSEGNANTI
DELLE SCUOLE SECONDARIE DI PRIMO GRADO

Joël Rivet, Didier Roy, Paolo Rossetti



Traduzione a cura di: Roberto Catanuto

SCRATCH

thymio

Cosa troverai in questo libro

Questo libro è scritto per gli insegnanti delle scuole secondarie di primo grado che desiderino sperimentare l'utilizzo del robot educativo Thymio durante le loro lezioni e laboratori di robotica educativa, programmandolo con il linguaggio Scratch e le sue estensioni per pilotare il robot.

Grazie al software Thymio Suite, disponibile gratuitamente sul sito thymio.org, si ha a disposizione un ambiente di programmazione con cui programmare Thymio utilizzando tutti i blocchi Scratch e avendo a disposizione altre dozzine di blocchi dedicati al robot Thymio senza la necessità di doversi collegare al sito Scratch del MIT.

Troverai una serie di attività didattiche ed educative organizzate in un crescendo di difficoltà per alunni dalle scuole secondarie di primo grado e la documentazione dei blocchi Thymio Scratch.

Dopo aver introdotto come allestire l'ambiente di programmazione Thymio Suite per programmare in Scratch, si chiarisce come avviene la comunicazione tra il robot e il linguaggio Scratch, vengono illustrati le tipologie di blocchi Thymio per Scratch e quindi si passa a descrivere venti attività didattiche e relative soluzioni.

Segue una rapida serie di consigli per gli insegnanti sulla costruzione di ostacoli per il robot e per la riproduzione di note musicali e si chiude con la documentazione dettagliata di ogni singolo blocco Scratch Thymio.

Sul sito del libro www.stripesedizioni.it puoi scaricare i codici delle attività didattiche in Scratch e le schede in formato PDF.

***Robotica educativa
ed intelligenza artificiale
Thymio e Scratch***

SCRATCH

Introduzione	2
Organizzazione dei laboratori con Thymio e Scratch	2
Thymio Suite: installazione e utilizzo	3
Particolarità della programmazione di Thymio con Scratch	5
Prerequisiti	5
Terminologia di programmazione	6
Tipologie di blocchi Scratch per Thymio	8
Legenda dei simboli che vi guideranno nella scelta delle attività	9
Tabella riassuntiva degli argomenti affrontati	10
Attività	12
Scopri i primi blocchi Scratch Thymio	12
Thymio al museo	13
Thymio segue la linea	14
Thymio sotto controllo	16
La staffetta Thymio	17
Il musicista	18
SOS Gas tossici	20
Thymio sta perdendo il controllo	21
Dobbiamo salvare il robot Susan (1)	22
Dobbiamo salvare il robot Susan (2)	24
Il gioco della bandiera	27
Il labirinto	28
Thymio sul banco di prova	29
Thymio scout	30
Thymio scientifico	31
Thymio equilibrista	33
Thymio ambulanza	34
Thymio al parcheggio	35
Avvio cifrato	36
Thymio impara a contare	37
Soluzioni delle attività	39
Consigli per l'insegnante	64
Blocchi Scratch per Thymio	66

Introduzione

Questa raccolta di attività su Thymio e Scratch è più destinata agli insegnanti o agli educatori che desiderano insegnare la programmazione con Thymio ai ragazzi dagli **11 ai 14 anni**. Il linguaggio utilizzato è Scratch del MIT al quale è stata aggiunta un'estensione per comunicare con il robot. Tuttavia, può essere adatta anche a chiunque sia interessato alla programmazione e all'insegnamento della robotica.

Il libro è composto da attività testate dagli insegnanti di una scuola secondaria di Bordeaux, Francia. Le attività sono progettate per per **sviluppare passo dopo passo il pensiero computazionale**, basato sulla programmazione di un oggetto tangibile come un robot, insegnando ai ragazzi a pensare in modo logico come farebbe uno sviluppatore di software. Le competenze così acquisite nell'organizzazione del pensiero possono essere applicate a molti settori: la scuola, i lavori intellettuali o manuali, oppure semplicemente nella vita quotidiana. Non possiamo dimenticare la dimensione ludica e il piacere degli alunni. Questo aspetto dovrebbe valere la metà dell'apprendimento stesso.

Organizzazione dei laboratori con Thymio e Scratch

La dimensione del gruppo dovrebbe essere dell'ordine di **16 studenti al massimo**. Gli studenti possono essere **abbinati a due a due, con un computer e un Thymio** per ogni coppia. Il numero di studenti nel gruppo non dovrebbe essere molto grande, per diversi motivi:

- pedagogico: l'aiuto personalizzato da dare diventa più oneroso quando aumenta il numero degli studenti, così come l'asimmetria tra chi si muove velocemente (e potrebbe annoiarsi) e gli altri.
- tecnico: se troppi Thymio senza fili sono collegati contemporaneamente, possono verificarsi interferenze nelle comunicazioni tra computer e robot, disturbando l'esecuzione dei programmi.

Le attività

La sezione "**La sfida...**" può essere proposta direttamente agli alunni. La sezione "**Aiuto alla programmazione**" può essere fornita o meno, così com'è, agli studenti. Nella maggior parte dei casi, si rivolge all'insegnante, che giudicherà le informazioni da comunicare in base alle conoscenze degli studenti, al tempo che si vuole dedicare all'attività e al livello di autonomia che si vuole lasciare agli studenti. È inoltre necessario dotare una stanza di spazio sufficiente, con un pavimento sul quale il Thymio possa muoversi senza ostacoli. Evitare pavimenti irregolari o piastrellati, soprattutto se leggermente sconnessi.

La **durata** prevista delle sessioni è di circa **50 minuti**, a volte anche meno. Ma a seconda delle circostanze come dimensioni del gruppo, velocità nello svolgimento delle missioni, problemi materiali o tecnici, proposte fantasiose da parte degli alunni, questo tempo può ovviamente cambiare. Inoltre, per evitare vari problemi che non mancano mai, è consigliabile anticipare e testare al meglio le sessioni.

Thymio Suite: installazione e utilizzo

Installazione

È richiesto un solo software, la Thymio Suite che si può scaricare da <https://www.thymio.org/it/programma/> scegliendo la versione corrispondente al proprio sistema operativo (**Windows, Mac o Linux**). Installare Thymio Suite o contattare l'amministratore di rete se non si dispone dei diritti necessari.

Iniziare a programmare

Se hai un Thymio collegato con cavo:

Utilizzare il cavo USB per collegare il robot ad una porta USB del computer. Il Thymio si accende automaticamente.

Se si dispone di un Thymio wireless con chiave USB:

Collegare la chiave ad una porta USB del computer e accendere il Thymio con il pulsante centrale. Anche per questo tipo di Thymio è possibile utilizzare una connessione cablata, soprattutto se si sospetta che la batteria del Thymio sia scarica.

Lanciare Thymio Suite.

Nella finestra che appare, cliccare sull'icona Scratch.



Thymio Suite: installazione e utilizzo

La nuova finestra dovrebbe mostrare l'immagine di un Thymio attivo.

Se i computer della classe sono collegati in rete, è possibile che tutti i Thymio collegati alla rete siano collegati appaiano sullo schermo di ogni computer della classe.

Dal momento che tutti i robot hanno lo stesso nome di default (Thymio II o Thymio II wireless), può diventare difficile distinguere i robot tra di loro e capire quale sia il proprio.

A seconda di quando viene lanciato Scratch, può accadere che un computer possa inavvertitamente prendere il controllo del robot di un altro computer.

Per evitare confusione, è possibile **rinominare** un robot. Questo nome sarà mantenuto nel robot stesso.

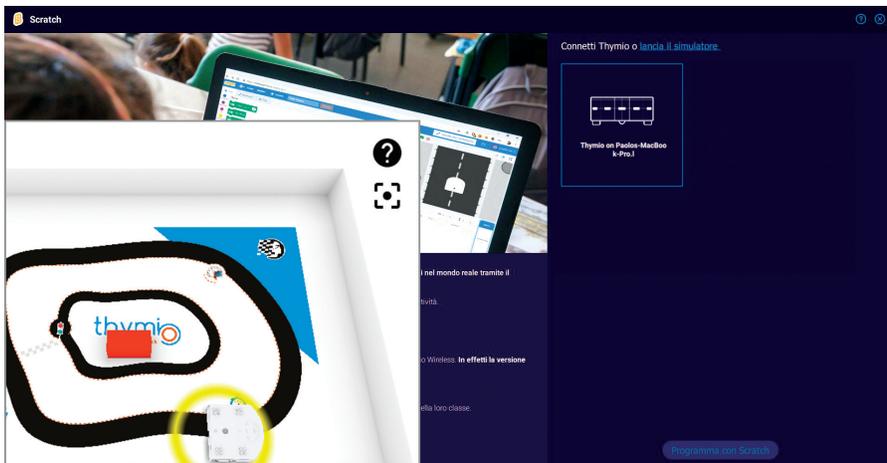
Per rinominare un robot, è necessario selezionarlo (una cornice azzurra circonda il robot), cliccare con il tasto destro del mouse e scegliere l'opzione Rinomina.

Per il nome, potrebbe essere interessante **scegliere il numero di serie del robot** (che andrà rinominato prima di lavorare con gli studenti).

Questo numero si trova sotto il robot. Ogni studente sarà in grado di identificare più facilmente il suo robot.

Per chi non riuscisse a collegare Thymio o non avesse il robot a disposizione Thymio Suite offre la possibilità di programmare tramite un simulatore. Per avviare il simulatore è necessario:

- **cliccare su lancia il simulatore**
- **scegliere un playground**
- **selezionare un Thymio (trasparente)**
- **cliccare su Programma con Scratch**



Particolarità della programmazione di Thymio con Scratch

Il processo di connessione tra Thymio e Scratch si differenzia dalla modalità normale.

Modalità normale

In modalità normale, utilizzata da VPL, Blockly o Aseba, la procedura è la seguente:

- L'utente crea il programma nell'editor dedicato (con l'apposita finestra).
- L'utente lancia il programma, e partono vari processi:
 - Il codice del programma viene compilato, cioè convertito in un codice macchina comprensibile dal computer interno al robot.
 - **questo codice macchina viene inviato** tramite il cavo o la chiave wireless **al robot che lo memorizza**.
 - il collegamento con il computer si interrompe, e Thymio esegue il programma in modo completamente indipendente. Questo è il motivo per cui il robot può essere scollegato e spostato dove si vuole.

Nota: tuttavia, se il filo rimane collegato o se la chiavetta è presente, Thymio ha la possibilità di inviare informazioni al computer, ma quest'ultimo non può intervenire sull'esecuzione del programma, se non per fermarlo.

Modalità Thymio e Scratch

Nella modalità che utilizza Scratch, la comunicazione non si interrompe dopo l'invio del programma al robot. Thymio invia le informazioni dai suoi sensori e riceve il codice da eseguire quando è necessario.

Questo è il motivo per cui **il collegamento tra Thymio e il computer non può essere interrotto, altrimenti il programma andrà in crash**.

Se si utilizza la modalità wireless con il tasto, il comportamento di Thymio può talvolta essere difettoso se le trasmissioni o le ricezioni delle onde sono disturbate.

Prerequisiti

Per trarre il massimo vantaggio da queste attività, è meglio:

- conoscere gli elementi essenziali di un robot come Thymio, cioè le nozioni di sensori e attuatori.
- Aver praticato uno degli altri 3 linguaggi (VPL, Blockly o Aseba) e aver assimilato la nozione di loop sensori-motorio specifico della robotica.
- Conoscere l'ambiente di programmazione Scratch e i blocchi principali (senza necessariamente padroneggiare tutti i dettagli).

Terminologia di programmazione

Per gli scopi di queste schede si tratta di chiarire il significato dato a certi termini nell'ambiente Scratch.

- **Blocco:** si tratta di forme geometriche trascinate nell'area del codice che possono innescare un'azione cliccando su di esse.
- **Programma:** rappresenta tutti i blocchi e le variabili presenti nell'area centrale del codice. Ciò corrisponde agli elementi che verranno salvati dal menu File / Salva sul vostro computer.
- **Script:** tutti i blocchi incollati insieme, o sotto un blocco di cappello come il blocco riportato in figura 1. Uno script è quindi l'equivalente di una funzione di gestione degli eventi come si può trovare in altri linguaggi. Un programma è semplicemente composto da uno o più script.
- **Codice:** termine generale che può designare in Scratch qualsiasi associazione di blocchi.
- **Sotto-programma:** è un blocco rosso appartenente alla voce i miei blocchi (vedi figura 2). L'utente può aggiungere blocchi normali sotto tale blocco. Nel programma principale si può usare questo piccolo modulo (vedi figura 3) per innescare l'esecuzione dell'intero blocco personalizzato.
- **Programma sequenziale:** i blocchi vengono eseguiti uno dopo l'altro, tenendo conto dei test **Se...allora** e dei loop. Un dato blocco viene eseguito solo quando il blocco precedente ha terminato la sua esecuzione. Questa è la forma più classica di programmazione, qui illustrata dall'esecuzione di un unico script.
- **Programmazione ad eventi:** in questo caso, un loop invisibile all'utente scansiona costantemente gli eventi che possono verificarsi: la pressione dei tasti, l'azione del mouse, l'arrivo delle misure del sensore ecc. Se il programmatore lo desidera, può prendere in considerazione, ad esempio con i blocchi cappello, un evento e attivare l'esecuzione di uno script in risposta a questo evento.
- **Programmazione concorrente:** in questo tipo di programmazione possono essere eseguiti contemporaneamente più script. Questa è la modalità di funzionamento naturale di un robot. Ad esempio, un robot dovrebbe essere in grado di avviare i suoi motori, accelerarli o fermarli mentre suona la musica o cambia i colori dei suoi LED.

La programmazione guidata dagli eventi genera naturalmente una programmazione concorrente.

(figura 1)

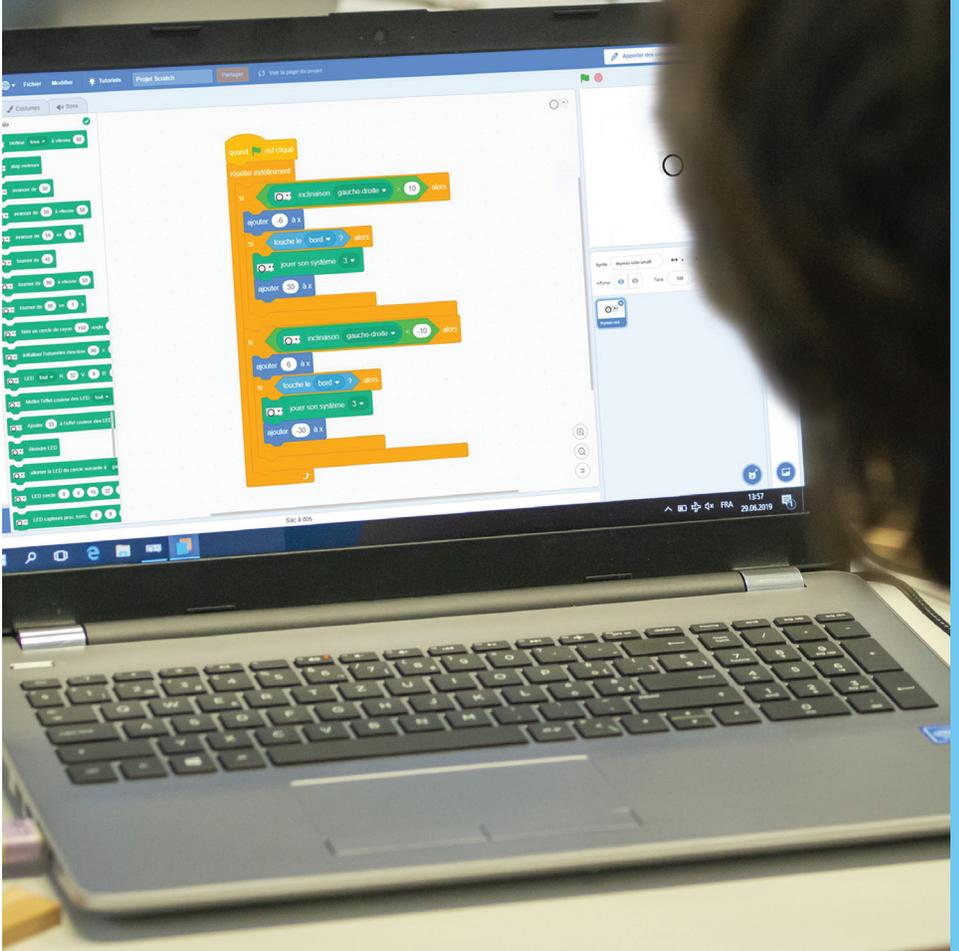


(figura 2)



(figura 3)





Tipologie di blocchi Scratch per Thymio

Al lancio di Scratch, Thymio Suite aggiunge un'importante serie di blocchi (ben 45!) che riguardano specificamente lo scambio di dati tra Thymio e Scratch.



Sono accessibili nella zona dei blocchi a sinistra in basso mediante questa icona. Come per gli altri blocchi Scratch, identifichiamo 4 famiglie grazie a alla loro forma:



I BLOCCHI ATTUATORI

Trasmettono un comando azione che Thymo deve eseguire.

Gli attuatori sono di 3 tipi:
movimento, luce, suono.



I BLOCCHI SENSORE

Con i **bordi arrotondati** che permettono di **inserirli in altri blocchi**, restituiscono le misure effettuate dal robot grazie a i suoi sensori.



I BLOCCHI LOGICI

Restituiscono il valore **VERO** o il valore **FALSO**.



I BLOCCHI EVENTI

Possiedono un **cappello**. Sono sempre all'inizio di uno script e sono attivi dal momento in cui compaiono nell'area script di Scratch.

Per la documentazione completa, fare riferimento alla documentazione alla fine del libro da pagina 66.

Convenzioni tipografiche

Nel testo un blocco è rappresentato in blu da simboli minore **<**, maggiore **>**, indipendentemente dalla tipologia.

- Le parentesi quadre **[]** indicano la presenza di un elenco di opzioni, separate dal simbolo **|**
- Le parentesi tonde **()** indicano un cerchio dove inserire un blocco sensore, una variabile o un numero intero relativo.

Esempio: il blocco qui rappresentato sarà descritto dall'istruzione:
< motore [tutti] con velocità (50) >



Legenda dei simboli che vi guideranno nella scelta delle attività in classe

In ogni scheda che descrive le attività di robotica educativa, utilizziamo i simboli qui sotto per aiutare l'insegnante nella preparazione delle attività educative, fornendo informazioni pratiche per le sessioni.



Durata indicativa delle attività in minuti. Questo è solo un'indicazione approssimativa.



Numero di robot Thymio necessari per svolgere l'attività.



Livello di difficoltà in scala da uno a tre. 1 si riferisce ad attività semplici e facili, 3 alle attività di più complesse e difficili.



Hai bisogno di altro materiale (carta, pennarelli, ecc.) per svolgere l'attività.



Per questa attività, sarebbe preferibile utilizzare un Thymio wireless invece di un robot collegato cavo USB.



Troverete altri documenti da scaricare sul sito web (codice di Scratch in formato **.sb3** ecc.).

Tabella riassuntiva degli argomenti affrontati

ORDINE	NOME	ELEMENTI DEL CODICE				
		DIFFICOLTÀ	NUMERO DI ROBOT	SENZA FILI	DOCUMENTO ONLINE	MATERIALE
1	Scopri i primi blocchi Scratch Thymio	1	1	X		
2	Thymio al museo	1	1	X		oggetti
3	Thymio segue la linea	1	1	X		linee nere
4	Thymio sotto controllo	1	1	X		
5	La staffetta Thymio	2	2 o più	X		linea nera
6	Il musicista	2	1			
7	SOS Gas tossici	1	1	X		carta A3, matita
8	Thymio sta perdendo il controllo	1	1	X	script .sb3	
9	Dobbiamo salvare il robot Susan (1)	1	2	X	script .sb3	disegno o oggetto
10	Dobbiamo salvare il robot Susan (2)	3	2	X	script .sb3	disegno o oggetto, cronometro
11	Il gioco della bandiera	2	2 o più	X	script .sb3	
12	Il labirinto	2	1	X		muri
13	Thymio sul banco di prova	2	1			
14	Thymio scout	2	1	X		muri, tetto, linea nera
15	Thymio scientifico	2	1	X	script .sb3	riga, carta o oggetto
16	Thymio equilibrista	2	1	X		superficie inclinata o pallone
17	Thymio ambulanza	2	2 o più	X		
18	Thymio al parcheggio	2	1	X	script .sb3	muri
19	Avvio cifrato	3	1			
20	Thymio impara a contare	3	1			indicatori

Tabella riassuntiva degli argomenti affrontati

		ELEMENTI DEL CODICE				
ORDINE	NOME	CICLO	CONDIZIONE	VARIABILE	PROG. CONCORRENTE	EVENTI SCRATCH
1	Scopri i primi blocchi Scratch Thymio		X			
2	Thymio al museo	X	X			
3	Thymio segue la linea	X	X			
4	Thymio sotto controllo				X	X
5	La staffetta Thymio					
6	Il musicista	X	X	X	X	
7	SOS Gas tossici					
8	Thymio sta perdendo il controllo	X	X	X		
9	Dobbiamo salvare il robot Susan (1)			X	X	X
10	Dobbiamo salvare il robot Susan (2)	X	X	X	X	X
11	Il gioco della bandiera		X		X	X
12	Il labirinto	X				X
13	Thymio sul banco di prova	X	X	X		
14	Thymio scout	X	X			
15	Thymio scientifico	X	X	X		X
16	Thymio equilibrista	X	X	X	X	X
17	Thymio ambulanza	X	X		X	
18	Thymio al parcheggio	X	X	X		X
19	Avvio cifrato		X	X		
20	Thymio impara a contare	X	X	X		

ATTIVITÀ S-01

Scopri i primi blocchi Scratch Thymio

Autori: **Joël Rivet, Didier Roy**



OBIETTIVI

- Scoprire i blocchi dei sensori di distanza, le luci e i motori
- Realizzare uno script con i primi blocchi Scratch
- Analizzare una situazione per modificare uno script Scratch

Scoperta di singoli blocchi

Fai scorrere il blocco < **pross. oriz. [centro]** > nell'area dello script. Clicca su di esso: viene visualizzato un numero.

Avvicina lentamente un dito al sensore centrale del Thymio e clicca di tanto in tanto sul blocco per vedere la modifica del valore visualizzato. Effettua il confronto con i LED rossi accanto ai sensori. Deduci una regola per il comportamento del blocco così come i possibili valori estremi. Testa anche le opzioni diverse da **[centro]**.

Fai scorrere nell'area dello script il blocco < **motore [tutti] con velocità (50)** > e cliccaci sopra: il robot si avvia (lascia la sorpresa agli studenti). Poi clicca sul pulsante di **arresto** Scratch. Chiedi agli studenti di trovare da soli il blocco che permette di fermare il robot.

Fai scorrere nell'area dello script il blocco < **LED [tutti] R(32) V(0) B(50)** > e cliccaci sopra. Prova diverse combinazioni di colori. Commenta se hanno visto la sintesi del colore RGB. Chiedi agli studenti di trovare da soli il blocco che spegne le luci.

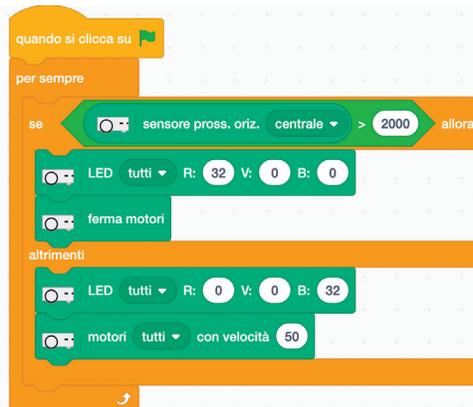
Primo script

Comporre lo script seguente, eseguirlo e descrivere ciò che fa.

Il veicolo autonomo

Se Thymio fosse un veicolo autonomo sulla strada, come dovrebbe reagire se un pedone gli passasse davanti? Chiedi la risposta agli studenti. (Dovrebbe fermarsi e poi ripartire una volta passato il pedone).

Una volta che tutti conoscono la risposta, modificare il programma aggiungendo due istruzioni in modo che Thymio si comporti come l'auto reale.



ATTIVITÀ S-02

Thymio al museo

Autori: **Joël Rivet, Didier Roy****OBIETTIVI**

- Analizzare un programma e scrivere in linguaggio naturale ciò che il robot fa
- Modificare un programma per raggiungere un obiettivo

Prerequisiti specifici: nessuno**La sfida: visitare il museo**

Il tuo Thymio sta visitando il Museo di Thymioville in fase di ricostruzione. Per il momento, ci sono statue e oggetti ovunque e Thymio non dovrebbe abatterli. Lo script qui sotto dovrebbe aiutarti.

Assembla il programma e descrivi per iscritto esattamente ciò che fa, utilizzando proposizioni condizionali: se ... allora

Lo script**Cambiamo il programma**

Può succedere che una statua decida di passare davanti a te. Devi quindi fermarti per lasciarla passare e riprendere quando hai spazio libero. Modifica lo script per far passare la statua!

```

quando si clicca su [bandierina]
  per sempre
    se [sensore pross. oriz. centrale] < 1000 allora
      motori tutti con velocità 50
    se [sensore pross. oriz. sinistra] > 2000 allora
      motori sinistro con velocità 75
      motori destro con velocità 0
    se [sensore pross. oriz. destra] > 2000 allora
      motori sinistro con velocità 0
      motori destro con velocità 75
    altrimenti
      motori sinistro con velocità -75
      motori destro con velocità -10
  
```

ATTIVITÀ S-03

Thymio segue la linea

Autori: **Joël Rivet, Didier Roy**

OBIETTIVI

- Scoprire i blocchi dei sensori del terreno e di distanza
- Elaborare una soluzione per guidare Thymio su una linea

Prerequisiti specifici: nessuno

La sfida: guidare Thymio come un treno

Perché i treni hanno i binari? Perché permettono di andare molto più velocemente senza deragliare. I binari fungono da guida molto efficace.

Metteremo Thymio sui binari ma gli daremo invece una linea nera da seguire. Questo accorgimento vale da tempo per i veicoli automatici in alcune grandi aziende.

Thymio ha due sensori del terreno nella parte anteriore per questo scopo.

Aiuto alla programmazione: Prova dei sensori di suolo

1 - Collocare Thymio su un foglio di carta bianca o su una superficie trasparente.

Far scorrere il blocco **< sensore terreno [sinistra] >** nell'area di programmazione. Cliccare su di esso: viene visualizzato un numero. Cambiare da **sinistra** a **destra** e cliccare di nuovo. Dovresti ottenere un valore molto vicino.

2 - Ripetere le misurazioni ponendo Thymio su una linea nera (vedi pag 64 - Consigli per l'insegnante - Realizzare linee nere).

3 - Ripetere le misurazioni ponendo Thymio su un foglio di carta bianca e sollevandolo da terra, poi girandolo verso l'alto.

Spiegazione fisica

Su un foglio di carta bianca, Thymio invia un raggio a infrarossi che viene riflesso dal foglio bianco e ritorna al sensore. Per questo motivo si ottiene un numero elevato.

Quando il raggio inviato da Thymio incontra una linea nera, viene assorbito e solo una piccolissima parte torna indietro. La misura indica un numero piccolo. Il sensore darà 0 solo in presenza di un nero perfetto (non facile da trovare!).

Allo stesso modo un Thymio rivolto verso l'alto invia un raggio che non torna indietro.

Ragioniamo con i sensori

Dobbiamo sviluppare un programma per tenere Thymio in linea mentre andiamo avanti.

All'inizio Thymio è sulla linea nera e parte ma non va proprio dritto e può girare leggermente a sinistra (o a destra).

- Da quale parte deve poi voltarsi per rimanere in linea?

- Quale sensore del terreno (sinistro o destro) cambierà il suo valore se Thymio gira leggermente a sinistra?

ATTIVITÀ S-03

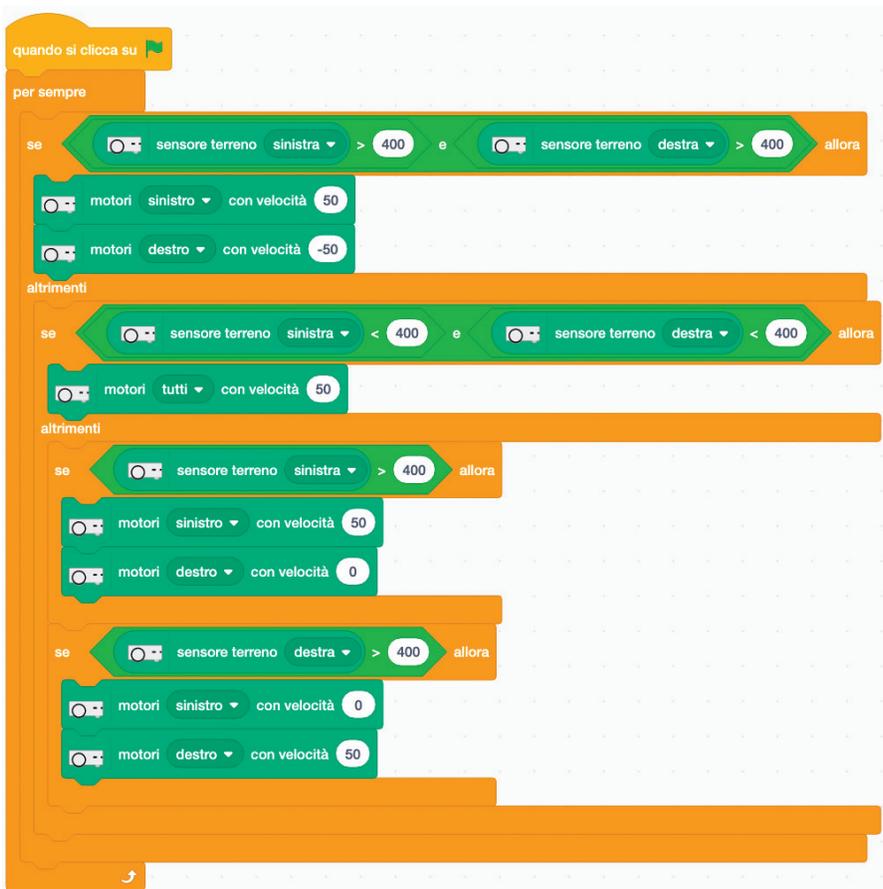
Thymio segue la linea

Con queste informazioni, effettua i test e dai ordini ai motori giusti con i blocchi Scratch.

- Non dimenticare anche di decidere come Thymio deve comportarsi se entrambi i sensori sono sulla linea nera.

Alla fine della linea, Thymio sarà sul bianco: potrà quindi fermarsi o girare su se stesso per riprendere il percorso. Se esce dalla linea nera di lato, si può girare su se stesso per tornare sulla traiettoria.

Provate tutti i tipi di linee, con curve più o meno strette.



The image shows a Scratch script designed for a Thymio robot to follow a black line. The script is contained within a 'per sempre' (forever) loop, which is triggered by a 'quando si clicca su' (when clicked) event. The logic is as follows:

- Initial State:** Both left and right ground sensors are set to a value of 400.
- When on the line (both sensors > 400):** The left motor is set to a velocity of 50, and the right motor is set to a velocity of -50, causing the robot to turn right.
- When off the line (both sensors < 400):** Both the left and right motors are set to a velocity of 50, causing the robot to turn left.
- When on the left side (left sensor > 400):** The left motor is set to a velocity of 50, and the right motor is set to a velocity of 0, causing the robot to turn left.
- When on the right side (right sensor > 400):** The left motor is set to a velocity of 0, and the right motor is set to a velocity of 50, causing the robot to turn right.

ATTIVITÀ S-04

Thymio sotto controllo

Autori: **Joël Rivet, Didier Roy****OBIETTIVI**

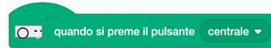
- Scoprire i blocchi degli eventi
- Scoprire la programmazione concorrente
- Implementazione di eventi Scratch con Thymio

La sfida: controllare i movimenti di Thymio

Si tratta di controllare i movimenti di Thymio con i suoi pulsanti o con la tastiera del computer.

Aiuto alla programmazione**I blocchi Thymio e il blocco a forma di cappello.**

Thymio ha 5 pulsanti sul dorso. Un pulsante centrale circondato da 4 tasti freccia. Possiamo usarli come vogliamo con i blocchi a forma di cappello. Possiamo aggiungere qualsiasi blocco sottostante. Non appena si preme il pulsante centrale, viene eseguito lo script sottostante. Non è nemmeno necessario avviare il programma con la bandiera verde. Ci sono 5 opzioni nel blocco, corrispondenti ai 5 pulsanti.

**Controllo dei movimenti di Thymio.**

Utilizzando il blocco precedente, crea un programma per Thymio che faccia le azioni seguenti:

- avanza alla velocità di 50 se si preme il pulsante anteriore
- indietreggia alla velocità di 50 se si preme il pulsante posteriore
- si gira a sinistra se si preme il tasto sinistro
- si gira a destra se si preme il tasto destro
- si ferma quando si preme il pulsante centrale.

Per divertimento, si può mettere Thymio su un pezzo di carta e un pennarello nel foro centrale di Thymio per disegnare una lettera quando il robot si muove.

Uso della tastiera

Esplora i blocchi nella sezione "Situazioni" sul lato sinistro di Scratch e trova il blocco che riguarda i tasti del computer. Sostituire i blocchi pulsanti del programma precedente con il blocco tasti. Prendi i tasti freccia della tastiera al posto dei 4 pulsanti del robot e il tasto spazio al posto del tasto centrale. Avete un nuovo modo di controllare Thymio! Vedi se è più facile disegnare una lettera in questo modo.

Fare uno scherzo ad un amico.

È necessario realizzare un programma di controllo con i 2 metodi tastiera e pulsanti. (10 script). Spiega a un amico che non conosce Thymio-Scratch come controllare Thymio con i pulsanti e lascialo giocare. Poi con la tastiera, digitate il contrario di quello che fa lui/lei. Dovrebbe metterlo un po' in difficoltà.

ATTIVITÀ S-05

La staffetta Thymio

Autori: **Joël Rivet, Didier Roy**



OBIETTIVI

- Sviluppare una soluzione senza aiuto esterno collaborando

Prerequisiti specifici: nessuno

La sfida: far collaborare i Thymio in una staffetta

“Vogliamo far percorrere una lunga distanza ai nostri Thymio, ma i robot sono stanchi. Non riescono a percorrere una distanza così lunga. Come si arriva alla meta?”

Aiuto alla programmazione

L'idea è di utilizzarne alcuni in staffetta, come nelle gare podistiche, per raggiungere il traguardo. Quindi c'è un Thymio alla linea di partenza e gli altri sono equamente distribuiti tra l'inizio e il traguardo.

Non riceverete alcuna assistenza. Se siete più squadre con un Thymio, è nel vostro interesse collaborare in modo che ognuna di esse possa portare un'idea ed elaborare una soluzione comune. Dovrete scoprire come un Thymio può aiutare un altro ad andare avanti.

Nota per l'insegnante

Il trucco è quello di utilizzare sensori posteriori che non sono spesso usati (il loro blocco Scratch è lo stesso di quello dei sensori centrali o laterali). Il traguardo sarà segnato da una linea nera. Sarà quindi necessario utilizzare il sensore centrale, i sensori posteriori destro e sinistro, i sensori del terreno.

Per le prove possono essere utilizzati i seguenti blocchi **< ripetere fino a quando <> >** o **i blocchi < ripetere fino a <> >**.

È anche importante che gli studenti comprendano che i diversi Thymio possono utilizzare lo stesso programma.

Se ogni squadra produce uno script diverso, può essere istruttivo incoraggiarli a fare un programma comune insieme.

Inoltre, è meglio mantenere le distanze tra i robot abbastanza brevi, perché questi non vanno mai veramente dritti.

ATTIVITÀ S-06

Il musicista

Autori: **Joël Rivet, Didier Roy**



OBIETTIVI

- Comprendere le particolarità della programmazione ad eventi
- Scopri le note e le frequenze

Prerequisiti specifici: nessuno

La sfida: musica con Thymio

Con gli strumenti musicali, ci sono diversi modi per suonare le note: con le chiavi, le fessure, i fori, le corde ... Con Thymio, è possibile utilizzare i 5 sensori orizzontali frontali per suonare le note sol, la, si, do, re ad esempio.

Ad una nota assegneremo anche un colore. L'idea è quella di far suonare una nota di durata fissa (diciamo 0,5 secondi), quando un sensore rileva un ostacolo, come ad esempio un dito.

Aiuto alla programmazione

Una prima nota

Per cominciare, scriviamo uno script che produce un SOL e accende i LED in blu usiamo il sensore di sinistra.

- Utilizza il blocco **< Aspetta fino a quando <>>**
- Cerca il blocco che permette di suonare la nota sol (vedi pag. 65 - Consigli per l'insegnante) per 0,5 s.
- Aggiungi un blocco per avere il colore blu.
- Aggiungi un blocco per spegnere la luce quando la nota è finita.
- Poiché la luce deve essere accesa per 0,5s, aggiungi un blocco per mantenerla in questa modalità: **< attendere (0,5) secondi >**.

Esegui lo script e avvicinati con un dito. Si sente il SOL. Ma se ti avvicini di nuovo con il dito, non funziona! Cerca di capire perché e come risolvere il problema.

Risposta. Devi mettere i blocchi in un ciclo del tipo **< ripetere all'infinito >**.

Lo strumento completo

Basta duplicare lo script (click destro) e cambiare i sensori, le note e i colori.

Si può adottare il codice delle note di colore: dalla nota più bassa alla più alta, mettere i colori blu, ciano, verde, giallo e rosso.

Nota per l'insegnante

È importante mettere in rilievo la differenza tra cliccare su un blocco del tipo **< quando si clicca la bandiera verde >** e l'icona della bandiera verde nell'angolo in alto a destra (in que-

ATTIVITÀ 5-06

Il musicista

sto caso, tutti gli script vengono eseguiti contemporaneamente). Quindi ci sono 5 loop in attesa in esecuzione allo stesso tempo.

- Allo stesso modo, il blocco **< aspetta fino a quando >** non è altro che un blocco **< ripetere fino a quando >** con niente dentro.

- In questo modo, gli script vengono eseguiti simultaneamente, quindi una nota può essere tagliata da un'altra, poiché Thymio può suonare solo una nota alla volta.

Come estensione, si può creare un blocco per nota e quindi scrivere uno script principale che suoni una melodia senza le dita (vedi soluzione 2).



ATTIVITÀ S-07

SOS Gas tossici

Autore: **Joël Rivet**

OBIETTIVI

- Dimostrare immaginazione e capacità di analisi
- Organizzare bene una sequenza di azioni
- Creare blocchi personalizzati

Prerequisiti specifici: nessuno

Materiale

Carta bianca (almeno formato A3), penna da inserire nel foro di Thymio.

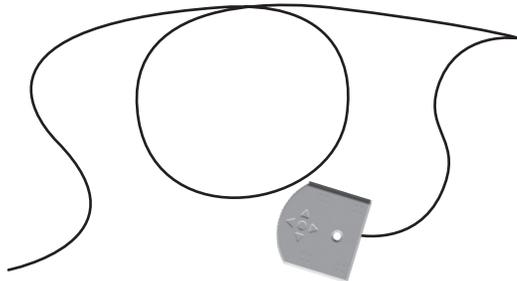
La sfida: scrivere SOS con Thymio

"Sono bloccato nell'edificio del laboratorio di robotica, non posso uscire a causa del gas tossico che si sta diffondendo! C'è stato un incidente industriale, la fabbrica accanto è saltata in aria. Se l'avessi saputo, avrei lasciato il posto come tutti gli altri alle 17:00. Ma no! Sono dovuto restare e lavorare sodo sul gigantesco robot Thymio che abbiamo finito di costruire ieri. Devo trovare un modo per segnalare la mia presenza senza lasciare l'edificio...

Ho un'idea: manderò il robot a tracciare un SOS sul pavimento. Userò il gigantesco Thymio e un grande pennarello per dipingere. I soccorsi probabilmente manderanno un drone sopra l'area, dovrebbe vedere il mio disegno. Il gigantesco Thymio può essere programmato proprio come i piccoli Thymio. Così posso testare il mio programma su un foglio di carta prima di inviare il robot. Sarebbe bello se il robot potesse anche fare più rumore e luce possibile. Non mi credi che esista il Thymio Gigante? Dai, guarda la foto sul sito web del libro. Dopo, mettiamoci al lavoro...!"

Nota per l'insegnante

Bisogna esercitarsi a disegnare lettere che si susseguono senza sollevare il pennarello. Le lettere possono essere unite da una linea orizzontale nella parte superiore o inferiore. Realizzare forme semplici: linee rette o archi di cerchio. Questa è un'opportunità per scoprire la creazione di blocchi (che si basano sulla nozione di procedura di programmazione imperativa). In ogni caso, il disegno non sarà molto preciso, ma non importa... molto meglio del gas tossico! Per aggiungere un po' di realismo, far eseguire la linea all'aria aperta e far volare un piccolo drone che scatterà foto.



ATTIVITÀ 5-08

Thymio sta perdendo il controllo

Autore: **Joël Rivet**

OBIETTIVI

- Conversione di uno script descritto in linguaggio naturale per rifarlo in un linguaggio di programmazione
- Structurare un programma

Prerequisiti specifici:

Conoscere le variabili Scratch

La sfida:

Per girare un film vi viene chiesto di simulare un attacco di follia di un robot. La sceneggiatura del film descrive la scena da costruire:

“All’inizio Thymio era calmo, poi ha iniziato a girare sempre più velocemente e gradualmente è diventato tutto rosso. Poi, alla massima velocità, ha continuato a girare per qualche secondo, emettendo un suono spaventoso. Ha rallentato gradualmente senza cambiare colore e si è fermato, ma c’era ancora un po’ di rumore come se ci fosse ancora tensione nei motori. Dopo 2 secondi ha fatto un altro po’ di rumore e si è fermato: niente più luce, niente più suono. Sembrava che fosse morto!”

Tutti i blocchi necessari sono forniti, ma sono sparsi in disordine. Il vostro compito è quindi quello di costruire il programma per replicare lo scenario.

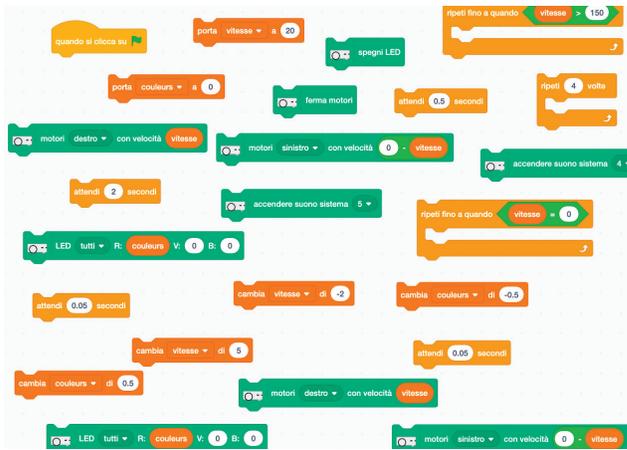
Note per l'insegnante

Il programma prevede tre fasi.

Nella prima, sarebbe interessante suggerire agli studenti di raggruppare i blocchi a tappe:

- 1 - Thymio accelera,
- 2 - Thymio urla,
- 3 - Thymio rallenta e si ferma,
- 4 - Thymio si spegne.

Scaricare il file di Scratch al seguente indirizzo: www.stripesedizioni.it/roboticaeducativa



ATTIVITÀ S-09

Dobbiamo salvare il robot Susan (1)

Autore: **Joël Rivet**

OBIETTIVI

- Implementare una soluzione concreta per un problema di sicurezza
- Scoprire la programmazione concorrente
- Utilizzo di eventi scratch per controllare il robot

Prerequisiti specifici: nessuno

La sfida: salvare il robot Susan

Thymio deve essere programmato per controllare a distanza il suo movimento. Deve trainare un altro robot guasto (Susan) alla base.

La storia

Siamo su Marte nell'anno 2055. Ben riparati nell'imponente edificio della SSEM (Stazione Scientifica di Esplorazione di Marte), da qualche ora si assiste a una violenta tempesta di sabbia che oscura l'orizzonte e scurisce il cielo marziano rosso pallido. La vostra attenzione è catturata da uno dei vostri robot, Susan (così si chiama) in difficoltà, a cento metri dalla vostra base. Purtroppo, le vostre telecamere di sorveglianza nel deserto marziano non mostrano alcun segno di stasi nella tempesta. Se non si fa nulla, la dannata sabbia finirà per penetrare nel funzionamento del nostro robot! Il tempo sta per scadere.

A quanto pare, le sue riserve di energia non sono esaurite, ma il robot Susan sembra incapace di raggiungere la base da sola. Almeno questo è indicato dai segnali di soccorso verdi che sta inviando automaticamente da diversi minuti.

Dovremo usare l'ultima risorsa: mandare un altro robot a prenderlo. Fortunatamente, tutti i robot che sono fuori uso hanno comunque la capacità di seguire una massa che si avvicina a loro. Una specie di traino senza cavo. A condizione che abbiano ancora abbastanza potenza e che il guasto, naturalmente, non influisca su un motore. Il vostro robot di salvataggio, Asimov (così si chiama) è nella camera di decompressione, pronto a uscire.

Realizzazione della scena

Dovrebbe essere sufficiente una superficie di 2m x 2m. Se ci limitiamo all'essenziale: è necessario un altro Thymio per simulare il robot rotto. Lo chiameremo Susan.

- Mettere il robot Susan in modalità verde preprogrammata (amichevole).
- Delimitare un'area a terra che rappresenta la camera di decompressione SSEM.
- Posizionare Asimov nella camera di equilibrio.
- Impostare un limite di tempo per il salvataggio (diciamo 1 minuto e 30). Fornire un cronometro.

ATTIVITÀ 5-09

Dobbiamo salvare il robot Susan (1)

Aiuto alla programmazione di Asimov

L'assistenza agli studenti potrebbe essere divisa in due parti:

1 - per mostrare (o ricordare loro) cosa si può fare con l'evento Scratch.

L'idea è quella di fargli indovinare che bisogna usare le frecce della tastiera per controllare i movimenti di Asimov.

- Freccia in alto per andare avanti.
- Freccia in giù per andare all'indietro.
- Freccia destra per girare a destra (sul posto) impostando le velocità delle ruote opposte.
- Freccia a sinistra per girare a sinistra.
- Un altro tasto (spazio per esempio) per fermare il robot.

quando si preme il tasto spazio ▾

2 - Utilizzare cinque blocchi cappello per ogni movimento. Quindi dovrete creare cinque script. Il cappello principale (bandiera verde) farà partire il robot.

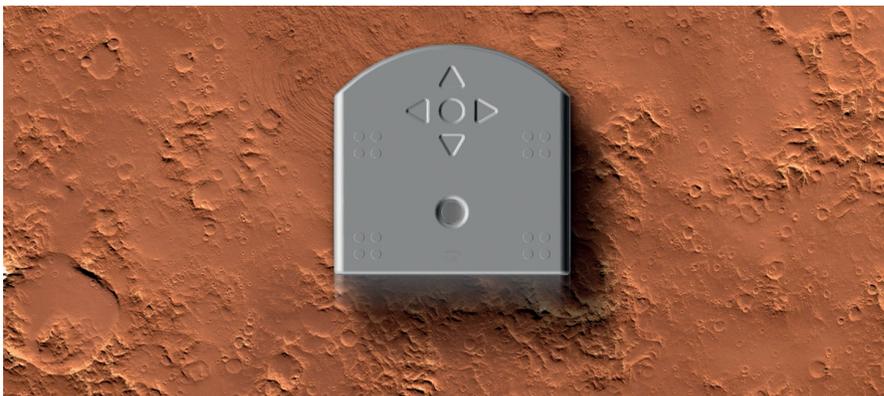
Può essere interessante utilizzare una variabile di velocità, la stessa per tutti gli script.

In questo modo sarà più facile regolare la velocità.

Per prolungare l'attività

A seconda di quanto tempo si decide di dedicare a questa attività, la si può arricchire:

- Fare squadre di 2 studenti e una gara tra le squadre per determinare il più veloce.
- Oppure far lavorare insieme le squadre per mantenere il tempo cumulativo il più breve possibile.
- Posizionare un ostacolo nel percorso per aumentare la difficoltà.
- Posizionare 2 Susan e 2 Asimov nella stessa area di gioco.
- Se vengono posizionati degli ostacoli, può essere assegnato un secondo Asimov per spingere gli ostacoli e sgombrare la strada per il traino.
- Utilizzare il vero programma di Susan implementato dall'insegnante.
- Fare decorazioni e vestire i robot.



ATTIVITÀ S-10

Dobbiamo salvare il robot Susan (2)

Autore: **Joël Rivet**

OBIETTIVI

- Implementazione di complesse regole di comportamento in uno script
- Scoprire la nozione di variabile di stato

Prerequisiti specifici:

Completare l'attività: Dobbiamo salvare il robot Susan (1)

Padroneggiare l'uso delle variabili

Padroneggiare il blocco (cronometro) del linguaggio Scratch

La sfida: programmare il comportamento di un robot in panne

Il contesto è descritto in modo esauriente nell'attività precedente.

Elenco dei compiti che devono essere eseguiti dal robot Susan

- Seguire il robot Asimov per vedere come realizza il comportamento "verde" pre-programmato.
- Fermarsi quando entra nella stanza di sicurezza (contrassegnata da una linea nera).
- Gestire la fase di risoluzione dei problemi in cui può seguire Asimov.

Durante questa fase deve:

- fare il conto alla rovescia verso il suo blocco definitivo e fermarsi
- accendere i suoi LED in verde lampeggiante ogni 2 secondi
- Gestione della fase di spegnimento:
 - ripetere 5 volte il suono di sistema 5 con il rosso lampeggiante, una volta al secondo
 - arrestare i motori, spegnere il maggior numero possibile di LED e riprodurre il suono di sistema 1.

Nota per l'insegnante

Il programma da scrivere è piuttosto complesso. Tuttavia, alcuni studenti che trovano "troppo facile" la programmazione di Asimov vorranno affrontare la programmazione di Susan. Ci sono molti modi per aiutarli.

Aiuto alla programmazione

Abbiamo diviso lo script in due parti, una che gestisce il tracking di Asimov e l'altra i parametri di comportamento (suono, luce, guasto).

Entrambi gli script sono lanciati contemporaneamente dal classico cappello < **quando si clicca la bandiera verde** >. Viene creata una variabile **guasto**. Poi viene inizializzata a 0 in entrambi gli script. Così, il valore 0 sarà usato all'inizio dei 2 script.

Strategia 1

Questa è la più semplice: fornire lo script scritto in italiano (vedi sezione soluzioni) e chiedere

ATTIVITÀ 5-10

Dobbiamo salvare il robot Susan (2)

di assemblare i blocchi seguendo il testo.

Strategia 2

Creare delle squadre (di 1 o 2 studenti), ognuna responsabile della scrittura di una parte dello script.

Parte "Seguire Asimov":

È la più complessa. La sua struttura logica è la seguente:

Inizializzazione

Spegnere i LED e impostare la variabile **velocità** su 50

Ciclo principale

Ripetere fino a guasto = 1

- se non c'è niente davanti al robot o se Thymio supera la linea nera, fermate i motori.
- oppure:
 - se c'è qualcosa davanti, andare dritto alla velocità **velocità** (Asimov è davanti, noi lo seguiamo)
- oppure:
 - se il sensore di sinistra rileva qualcosa (Asimov è sulla sinistra), girare a sinistra.
 - se il sensore di destra capta qualcosa (Asimov è sulla destra), girare a destra

Nota

Se Scratch si blocca, metti un blocco **<aspetta (0.1) secondi>** nel ciclo.

Parte gestione comportamentale

Può essere condivisa tra 2 Team:

- Il Team 1 si occupa dell'inizializzazione e della fase di risoluzione dei problemi dove guasto = 0.
- Il Team 2 si occupa del resto dove il guasto passa a 1, e questa è la fase di stop.

Struttura della fase di risoluzione dei problemi

- inizializzare guasto a zero e aggiungere una variabile **tempo_totale** che definisce il tempo di vita del robot. Azzerare il cronometro.
- ripetere fino a quando il cronometro non supera il **tempo_totale**
far lampeggiare i LED in verde in cicli di 2 secondi

Struttura della fase di spegnimento

- Il Thymio non risponde più ai valori del sensore.
- Ripetere 5 volte.
- Riprodurre il suono di sistema 5.
- Far lampeggiare i LED di colore rosso in cicli di 1 secondo.
- Spegnere i LED e i motori.
- Riprodurre il suono di sistema 1.

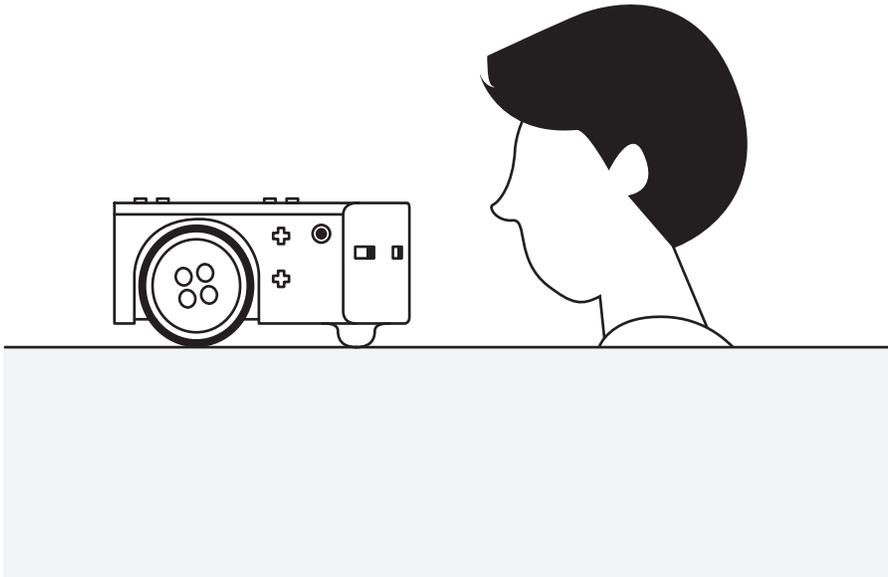
ATTIVITÀ 5-10

Dobbiamo salvare il robot Susan (2)

Nota su Asimov e Susan

Non conosci Susan? Normale, non è un robot, ma il nome del primo robot-psicologo. La dottoressa Susan Calvin appare nei primi romanzi di fantascienza di Isaac Asimov. Asimov è stato il primo scrittore a sviluppare storie di robot basate sulla logica e non sulla fantasia. Secondo l'Oxford English Dictionary, ha persino inventato la parola "robotica". E ha creato, con John W. Campbell (1940) le 3 leggi della robotica, che sono le seguenti:

- 1.** un robot non deve danneggiare un essere umano, né, rimanendo passivo, permettere che un essere umano venga esposto al pericolo;
 - 2.** un robot deve obbedire agli ordini impartiti da un essere umano, a meno che tali ordini non siano in conflitto con la prima legge;
 - 3.** un robot deve proteggere la sua esistenza purché tale protezione non sia in contrasto con la prima o la seconda legge.
- Va da sé che queste regole sono troppo complesse per il nostro Thymio, e in effetti per ogni robot costruito sul pianeta.



ATTIVITÀ 5-11

Il gioco della bandiera

Autore: **Joël Rivet**

OBIETTIVI

- Controllare i movimenti di Thymio con la tastiera a scopo ludico

Prerequisiti specifici: nessuno

La sfida: Il gioco della bandiera (o ruba-bandiera)

Due avversari sono faccia a faccia. Ognuno di loro ha una bandiera sulla schiena, che si infila nella cintura o nei pantaloni. I due avversari si girano per afferrare la bandiera dell'altro. Il primo che ci riesce vince.

Adattiamo questo gioco a 2 Thymio. Sostituiamo la bandiera con i sensori posteriori. Il Thymio che ha i sensori posteriori accesi ha perso. Bisogna stare in uno spazio aperto, in modo che il Thymio non vada a sbattere in qualcosa. I 2 Thymio sono controllati dai tasti freccia sulla tastiera da 2 studenti. Se uno studente vuole giocare da solo, uno dei robot può essere posizionato nella modalità rosso (timido) pre-programmata.

Aiuto alla programmazione

Il programma sarà identico per entrambi i Thymio. Sono necessari 5 script per gestire il movimento di Thymio tramite i tasti della tastiera (vedi attività 4 o 9), uno script principale che fa lampeggiare di rosso il Thymio perdente per qualche istante. È utile colorare anche i 2 Thymio per distinguerli (evitare il rosso). È anche possibile aggiungere suoni.

Per prolungare l'attività

Per gli studenti che già sapevano come programmare i tasti (attività 4 o 9), è possibile migliorare il controllo giocando con la variazione di velocità. Il principio è quello di creare e impostare una variabile di velocità iniziale a zero (all'inizio dello script principale), poi la freccia in alto aggiunge un valore, ad esempio 5, alla velocità ad ogni pressione, e la freccia in basso rimuove questo valore. Per le rotazioni, sostituire i valori fissi con la variabile **velocità e velocità/4 o 0**. È possibile mettere più avversari. Un Thymio che ha "perso la bandiera" è fuori gioco, finché non ne rimane uno solo.



ATTIVITÀ S-12

Il labirinto

Autore: **Joël Rivet**

OBIETTIVI

- Stabilire un semplice algoritmo di uscita dal labirinto
- Implementazione di un algoritmo Scratch

Prerequisiti particolari: nessuno

Perso in un labirinto?

Ci sono molti modi per uscire da qualsiasi labirinto. Immaginiamo un labirinto fatto di muri. Il metodo più semplice è quello di andare lungo un muro, a destra o a sinistra (ma sempre dallo stesso lato) senza mai lasciarlo. Funziona di sicuro a condizione che non ci sia un'isola (muri al centro non collegati ai muri ai bordi).

Aiuto alla programmazione

Stabilire l'algoritmo di uscita

Un buon metodo è quello di disegnare un piccolo labirinto sulla carta, posizionare un piccolo oggetto lungo un muro e poi fare un inventario di tutte le situazioni che può incontrare mentre avanza lungo le mura. Se decidiamo che Thymio vada lungo la parete sinistra, non preoccupiamoci della parete destra. Basta considerare cosa succede tra un muro e i sensori a sinistra e al centro. Elenca i movimenti di Thymio secondo il muro in linguaggio naturale. In questo modo si stabilisce l'algoritmo di progressione attraverso il labirinto fino all'uscita.

Lo script Scratch

Dall'algoritmo, assembla lo script. Pensa anche a fermare il programma al termine della missione. Inoltre, Thymio ha modo di sapere che è fuori dal labirinto?

Soluzione in linguaggio naturale

Se Thymio ha un muro davanti a sé, si gira a destra oppure: Se Thymio ha un muro a sinistra, va dritto

Se Thymio non ha un muro a sinistra, gira a sinistra.

La difficoltà è che Thymio non va mai perfettamente dritto. Quindi la frase: "Se Thymio ha un muro a sinistra, va dritto" va interpretata come segue: se Thymio vede la parete sinistra, gira leggermente a destra, altrimenti gira leggermente a sinistra. Certo, Thymio gira sempre leggermente, ma in questo modo si controlla il suo movimento un po' meglio.

Implementazione dello script.

Sarà necessario avere molti muri con passaggi non troppo stretti in modo che Thymio possa girare facilmente. Pertanto, è necessario fornire una superficie appropriata. Il labirinto scaricabile dal sito richiede 25 pareti (vedi pag 64 - Consigli per l'insegnante). Tutti i gruppi useranno lo stesso labirinto. Pianifica anche che se Thymio tocca un muro, questo rimarrà al suo posto senza muoversi all'indietro. Non avvicinare troppo i muri tra loro, questo renderebbe il movimento più complicato.

ATTIVITÀ 5-13

Thymio sul banco di prova

Autore: **Joël Rivet**

OBIETTIVI

- Utilizzo di variabili per i motori
- Aumentare una variabile

Prerequisiti specifici: conoscere le variabili di Scratch

La sfida: testare i motori di Thymio

I motori di Thymio sono in buone condizioni? Perché non li facciamo rombare un po'? Dobbiamo scrivere un programma che dia a Thymio il seguente comportamento:

- All'inizio è fermo
- Poi accelera, va sempre più veloce fino a quando non si ferma all'improvviso, per non spingere troppo il motore.

Aiuto alla programmazione

Crea una variabile **velocità** con un valore di 5 per iniziare.

Poiché Thymio deve andare sempre più veloce, la velocità deve aumentare poco a poco. Quale operazione può essere utilizzata per aumentare un numero? Aggiunta, sottrazione, moltiplicazione, divisione?

- Quindi trova nei blocchi del campo variabile nella colonna di sinistra di Scratch un blocco che può aumentare la variabile **velocità**. Basta fare una ripetizione in cui ogni volta si aumenta un po' la variabile **velocità**.
- Infine, mettere i motori di Thymio alla variabile **velocità** nel ciclo e il robot comincia ad accelerare.

Un'ultima cosa, come fermare Thymio? Scoprillo provando la velocità massima di Thymio, cioè la velocità oltre la quale Thymio non accelera più. Poi usa questa velocità in un test con il blocco di ripetizione che ritieni opportuno.

Per prolungare l'attività

Modificare il programma in modo che, invece di accelerare, Thymio rallenti dalla velocità massima fino a zero. Il principio dell'aumento di una variabile può essere applicato anche ad un suono che diventa sempre più alto (o più basso). Per l'aumento, utilizza la moltiplicazione e vedi la soluzione del programma 2. Possiamo anche aumentare la luce con lo stesso procedimento.

Osservazioni:

- Un aumento inferiore a 5 non avrà alcun effetto.
- Se l'accelerazione è troppo veloce, meglio aggiungere un blocco di attesa con un certo intervallo di tempo.

ATTIVITÀ S-14

Thymio scout

Autore: **Joël Rivet**

OBIETTIVI

- **Costruire uno script risolutivo senza aiuto e con la collaborazione del team**

Prerequisiti specifici: Saper seguire i muri o una linea nera.

La sfida: esplorare un tunnel

“Thymio è stato assunto dal municipio di Thymioville per esplorare una vecchia stretta galleria, troppo pericolosa per i lavoratori comunali. La sua missione è la seguente: deve entrare nel tunnel e attraversarlo per intero. Se incontra qualcosa, fa retromarcia con la luce rossa accesa e un suono, per indicare che ha incontrato un ostacolo, fino a quando non esce dall’ingresso. Se alla fine del tunnel non ha incontrato nulla, fa retromarcia allo stesso modo ma con luce verde e nessun suono”. La fine del tunnel sarà segnata da una linea nera.

Note per l’insegnante

La riflessione sulla soluzione da trovare per la programmazione di Thymio può essere oggetto di discussione tra tutti gli alunni. L’insegnante potrebbe dover intervenire per aiutare a formulare domande semplici:

- Quali sensori dovrebbero essere utilizzati? Identificare i blocchi corrispondenti.
- Quali test si dovrebbero fare?

Se gli studenti non hanno mai programmato come far seguire una parete al robot o delle linee nere (in Scratch o VPL), considerare la possibilità di dividere il programma in 2 parti.

1 - programmare Thymio per muoversi tra “muri” (abbastanza vicini tra loro)

Il fatto che le pareti siano vicine tra loro (1 cm tra Thymio e le pareti) rende necessario confrontare i valori dei sensori di destra e di sinistra (piuttosto che testarne uno solo) e impostare valori di rotazione morbida (motore sinistro 60 e motore destro 40 per esempio).

2 - programmare Thymio per innescare la retromarcia e reagire in modo diverso se incontra una linea nera o un ostacolo davanti ad essa.

L’idea è quella di arrivare (sulla lavagna per esempio) a una versione dello script in linguaggio naturale. Poi ogni gruppo implementa lo script in blocchi Scratch.

Implementazione fisica

Il tunnel può essere realizzato con normali pareti in cartone e il tetto con lastre leggermente più rigide. Per realizzare i muri vedi pag. 64 - Consigli per l’insegnante.

Diversi tunnel Ogni gruppo dovrebbe essere in grado di realizzare un tunnel di fortuna per testare il proprio script. Ma, ben lontano nella stanza, ci deve essere un tunnel fatto dall’insegnante, il cui contenuto non è visibile agli studenti. Come ostacolo, può essere messa nel tunnel una scatola di Thymio. Una volta che un gruppo ha realizzato e testato il suo script, viene a provarlo nel tunnel dell’insegnante.

ATTIVITÀ S-15

Thymio scientifico

Autore: **Joël Rivet**

OBIETTIVI

- Introduzione al metodo scientifico attraverso la misurazione della velocità
- Organizzare uno script da blocchi sparsi.

Prerequisiti particolari: nessuno

La sfida: misurare la velocità reale di Thymio

A Thymioville, in una bella mattina di sole, i Thymio hanno visto apparire per le strade i segnali dei limiti di velocità. Ma si sono subito accorti che il numero sul cartello (10 cm/s) non corrispondeva ai valori dei blocchi Scratch. Il Thymio, in fretta e furia, viene a cercare lo scienziato della città (che sei tu), per scoprire a quale valore corrisponde questo limite di velocità, "10 cm/s". Così si arriva al compito di misurare la velocità di Thymio nel miglior modo possibile per trovare la corrispondenza.

Aiuto alla programmazione

Il principio consiste nel misurare la distanza percorsa da Thymio in un determinato periodo di tempo. La distanza viene misurata con un metro e il tempo con il cronometro di Scratch.

Flusso del programma

È diviso in 3 parti:

- Preparare la misura chiedendo all'utente di marcare la posizione di partenza.
- Far muovere Thymio e registrare la distanza in cm e il tempo in secondi.
- Utilizzando queste 2 misurazioni, calcolare la velocità reale di Thymio con la formula: $\text{velocità} = \text{distanza} / \text{tempo}$. Completare e visualizzare il risultato.

I blocchi del programma sono forniti sul sito www.stripesedizioni.it/roboticaeducativa.

Aiuto sui blocchi Scratch

I blocchi di Scratch non sono tra i più utilizzati, ma c'è un aiuto qui sotto:

- Il blocco **< chiedere [] e aspettare >** consente di porre una domanda all'utente e di recuperare la risposta. Quando il programma è in esecuzione, l'utente deve solo digitare la sua risposta e convalidare. La risposta dell'utente è memorizzata nel blocco blu **< Risposta >**.
- I blocchi **< dire [] >** o **< pensa a [] >** mostrano solo un testo.
- Ci sono due blocchi per gestire il cronometro:
- Il blocco **< azzerare il cronometro >** che lo azzerava. Attenzione, contrariamente ad un vero cronometro, non c'è un pulsante per riavviarlo, si riavvia non appena viene inizializzato.
- Il blocco **< cronometro >** che fornisce il tempo in secondi trascorso dal reset.

Utilizzando tutte queste informazioni, scrivere il programma rimontando i blocchi.

Thymio le scientifique

Sfruttare i risultati

Si tratta di mettere in relazione il blocco di velocità, chiamato **Vbloc** che digitiamo in Scratch e la velocità sperimentale misurata e calcolata, chiamata **Vexp** (exp sta per sperimentale). Si devono quindi di eseguire più misurazioni con diverse velocità.

La tabella fornita indica una serie di valori e deve essere compilata dagli studenti.

Numero esperienza	1	2	3	4	5	6	7	8
Blocco Velocità	25	50	75	100	125	150	175	200
Velocità misurata								

Domande:

- 1- Qual è l'unità di misura della velocità **Vexp**?
- 2- Come si possono collegare i valori delle velocità **Vbloc** e **Vexp** per le prime colonne?
- 3- Cosa succede dopo? Qual è la velocità massima di Thymio?
- 4- Qual è la velocità limite da Thymio a Thymioville, espressa in velocità **Vbloc**?

Risposte:

- 1- **Vexp** è espresso in cm/s
- 2- In linea di principio, nella tabella si devono osservare due parti: dal 25 al 125, c'è una proporzionalità approssimativa tra le 2 velocità.
- 3- Sopra i 125, la velocità di Thymio è al massimo e non aumenta più, qualunque sia la velocità inserita nel blocco.
- 4- Il limite di velocità è di 100.

Nota per l'insegnante

Questa attività è pensata in particolare per gli studenti che sono attratti dalla scienza e dalla fisica in particolare. Non esitare a spiegare loro che stanno applicando un vero e proprio metodo scientifico, che dovrebbe migliorare il loro lavoro. Se hanno un insegnante di fisica, che i ragazzi non esitino a parlargli di questa attività. (L'autore di queste attività è anche un insegnante di fisica).

ATTIVITÀ S-16

Thymio equilibrista

Autore: **Joël Rivet**

OBIETTIVI

- Scoprire il sensore accelerometro
- Utilizzare le relazioni tra le variabili

La sfida: mantenere l'equilibrio su un piano inclinato

Thymio può mantenere l'equilibrio su una tavola inclinata o una grande palla rotolante? Come può controllare la sua velocità? Bisogna fare uno script che adatti la velocità all'inclinazione quando si sposta il supporto avanti e indietro.

Aiuto alla programmazione

Il blocco di inclinazione

Per misurare l'inclinazione, c'è un blocco chiamato **< inclinazione [fronte-retro] >**.

Il nome del dispositivo è un accelerometro.

È possibile testare il suo valore con il loop:

clickando qui: 

Il valore può essere visualizzato nella scena Scratch.



Lo script: Versione semplice

La logica sta in 3 frasi da completare:

Se Thymio è orizzontale, la velocità è fissata a: ...

Se Thymio è inclinato all'indietro, la velocità è fissata a: ...

Se Thymio è inclinato in avanti, la velocità è fissata a: ...

I valori migliori si possono trovare facendo dei test, e naturalmente, ci vorrà un ciclo.

Versione migliorata

Si può vedere che più ripida è la pendenza, più alta deve essere la velocità. Creiamo quindi una variabile chiamata **pendenza**, e un'altra variabile chiamata **velocità**. Cerca di trovare per tentativi un numero che possa collegare le 2 velocità.

I passi potrebbero essere:

- mettere in **pendenza** il valore dell'inclinazione
- impostare la **velocità** come prodotto della pendenza per numero.
- mettere **velocità** nei blocchi dei motori

Per prolungare l'attività

Possiamo mostrare l'evoluzione della pendenza nello script di Scratch.

Aggiungi l'estensione "Disegna usando gli sprite". Posiziona all'inizio del blocco **< penna in posizione di scrittura >**. Poi nel ciclo metti i blocchi **< avanzare di (5) passi >** e **< aggiungi (velocità) a y >**.

ATTIVITÀ S-17

Thymio ambulanza

Autore: **Joël Rivet**

OBIETTIVI

- Scoprire e utilizzare l'altoparlante
- Utilizzo della programmazione concorrente

Prerequisiti specifici: aver utilizzato diversi script concorrenti.

La sfida: Toglietevi di mezzo!

Thymio è stato trasformato in ambulanza, c'è un uomo ferito a bordo. Sta andando all'ospedale di Thymioville a tutta velocità, a sirene spiegate! Purtroppo, un taxi è lento a liberare la strada davanti a lui. Thymio è costretto a fermarsi. Ma al suono della sirena, il taxi finalmente accosta a destra, sgombrando la strada. Thymio può continuare il suo cammino.

Messa in scena

Sono necessari almeno 2 Thymio, che avranno 2 diversi programmi. Un Thymio avrà il ruolo di ambulanza, l'altro sarà il taxi in sosta. Se ci sono diversi altri Thymio, si possono sistemare in fila sul percorso dell'ambulanza come tanti taxi. In termini di squadre di studenti, solo una squadra può programmare l'ambulanza e le altre squadre gli altri veicoli, taxi ecc.

Aiuto alla programmazione

Comportamento del Thymio

L'ambulanza Thymio si sta muovendo a tutta velocità. Allo stesso tempo, emette il classico "eoooh eoooh" o qualsiasi altro suono di ambulanza. Quando incontra il taxi di fronte, si ferma, molto vicino al Thymio. Quando il taxi si sposta, ricomincia.

I taxi Thymio sono fermi. Non appena sentono la sirena, iniziano con una traiettoria curva verso destra.

Programmazione dell'ambulanza

Ci vorranno 2 Script:

- uno che fa il loop delle note RE - LA, per 0,7 s ciascuno
- l'altro che gestisce il movimento: velocità massima se non c'è niente davanti, stop se c'è un ostacolo, in un loop infinito.

Programmazione del taxi

Vedere la documentazione relativa ai blocchi alla fine del libro per il blocco associato al microfono. Indicazione per lo script: attendere che il microfono rilevi qualcosa, poi fare una curva a destra con il blocco **< fare un cerchio di raggio () angolo () >**, poi la stessa curva ma a sinistra.

ATTIVITÀ 5-18

Thymio al parcheggio

Autori: **Joël Rivet, Paolo Rossetti**

OBIETTIVI

- Comprendere e utilizzare un codice esterno creato da altri programmatori
- Costruire uno script a partire da codici esterni

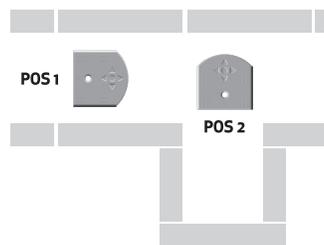
Prerequisiti specifici: nessuno

La sfida: parcheggiare una vettura autonoma

In attesa che le vetture autonome si diffondano, chiediamo a Thymio alcuni comportamenti caratteristici: spostarsi su una strada, parcheggiarsi da solo, uscire dal parcheggio e riprendere la sua strada. Per fare tutto questo, Thymio usa dei sensori anteriori e posteriori.

Messa in opera dello script

Posizionare dei muri come suggerisce l'immagine. Rispettare la lunghezza dei muri non è tuttavia indispensabile. La rientranza in basso rappresenta lo spazio di parcheggio. I muri possono essere realizzati con dei fogli di carta (vedi pag 64 - Consigli per l'insegnante).



Aiuto alla programmazione

Sottoprogrammi forniti

Per aiutarti, nel sito, vengono forniti degli script all'interno di due sotto-programmi:

- Sotto programma **tra_mura**. Ricopiare o caricare questo sottoprogramma mediante il blocco Scratch rosso apposito, mettere Thymio sulla strada (POS 1) ed eseguirlo. Descrivere quello che fa. Come si potrebbe arrestare la sua esecuzione?
- Sotto-programma **indietreggiare**: analogamente a prima, ricopiare o caricare questo sottoprogramma, mettere Thymio in (POS 2), e farlo eseguire. Descrivere quello che fa. Come si ferma Thymio? Aggiustare se necessario il valore del secondo blocco **< distanza(dietro) >**.

Il tuo script

Creare uno script utilizzando questi due sotto-programmi allo scopo che Thymio abbia il comportamento seguente: Thymio è all'inizio della strada, avanza evitando i muri fino a che si trova con l'area di parcheggio alla sua destra. Si preme sul tasto **spazio** della tastiera, quindi Thymio si mette in (POS 2) (rotazione di 1/4 di giro) e si parcheggia.

Adesso, premendo di nuovo il tasto **spazio**, Thymio esce dal parcheggio e riprende il suo cammino verso destra, evitando i muri. Per l'uscita dal parcheggio, ci si può accontentare di un blocco per avanzare della giusta lunghezza, allo scopo di ritrovarsi di nuovo sulla strada. Fare comunque dei tentativi per trovare lo spostamento corretto.

ATTIVITÀ S-19

Avvio cifrato

Autore: **Joël Rivet****OBIETTIVI**

- Progressi nella gestione degli eventi
- Utilizzare il connettore logico «OPPURE»
- Sviluppare l'iniziativa e l'immaginazione

La sfida: consentire a Thymio di avviarsi solo se si digita il codice giusto

A ThymioVille, il comune ha deciso di mettere Thymio a disposizione della popolazione. Per evitare abusi, il comune emette su richiesta un codice cifrato per il rilascio di un Thymio. Si tratta di toccare i tasti Thymio nell'ordine corretto. Dopo di che, si sblocca il robot. Se viene premuto un pulsante sbagliato, il Thymio diventa rosso ed emette il suono del robot quando viene fermato. All'inizio Thymio è blu e, quando comincia il programma, diventa verde.

Aiuto alla programmazione**Cosa diresti in italiano?**

Il problema deve essere affrontato in 4 fasi ad incastro. Ad ogni passo, aspettiamo che l'utente prema un pulsante. Se il pulsante è corretto, passiamo al passo successivo, altrimenti fermiamo lo script. In linguaggio naturale, la prova di un passo sarebbe simile a questa: "attendere finché non si preme il pulsante **anteriore** o il pulsante **posteriore** o il pulsante **sinistro** o il pulsante **destro**".

In Scratch

Il blocco Scratch Thymio **< pulsante [avanti] >** restituisce VERO nel momento in cui viene premuto il pulsante frontale del Thymio, restituisce FALSO altrimenti. In Scratch, la soluzione è usare il blocco **< aspetta fino a quando < condizione > >** con la condizione: **< pulsante [avant] > o < pulsante [indietro] > o < pulsante [sinistra] > o < pulsante [destra] >**. Una volta che questa condizione è stata soddisfatta, il passo successivo è quello di determinare se il pulsante corretto è stato premuto. Se lo è, inseriamo un altro controllo, altrimenti ci fermiamo. Un test se **< pulsante [...] altrimenti >** è sufficiente. Tra due fasi, è prudente aggiungere un piccolo tempo di attesa **<attendi (0,5) secondi>** per impedire che Scratch consideri contemporaneamente due blocchi **< aspettare fino a quando >**. I blocchi da usare in caso di errore del codice possono essere incollati in un blocco personalizzato che contenga le 2 parti **<emettere il suono di sistema (1)>** e **<LED [tutti] R(32) V(2) B(0)>**

Per prolungare l'attività

Una volta che gli studenti hanno padroneggiato la gestione degli eventi, possono essere autorizzati a creare la propria combinazione di tasti. Alcuni possono avere l'idea di usare le dita davanti ai sensori invece di digitare i tasti. Bene, non dobbiamo più toccare Thymio! Il codice è lo stesso, tranne che per la condizione di prova.

ATTIVITÀ 5-20

Thymio impara a contare

Autore: **Joël Rivet**

OBIETTIVI

- Programmazione avanzata: strutturare uno script con blocchi
- Programmazione avanzata: utilizzo di un loop con una variabile incrementale

Prerequisiti specifici:

Utilizzare variabili (lettura, assegnazione) e blocchi condizionali **< Se ... >**.

La sfida: insegnare a Thymio a contare con i suoi sensori

Messaggio agli studenti: “Da piccolo, hai imparato a contare con le tue cinque dita. Potresti insegnare a Thymio a contare con i suoi cinque sensori?”. L’idea è quella di posizionare cinque oggetti sottili (ma non troppo sottili) proprio di fronte ai sensori frontali, come i pennarelli della lavagna bianca. Lo script deve essere in grado di contare il numero di oggetti che si trovano di fronte e di farlo suonare tante volte.

Aiuto alla programmazione

Possiamo immaginare un script che utilizzi due blocchi personalizzati.

- Il blocco **< conteggio >** che produce un numero variabile durante la lettura dei sensori.
- Il blocco **< dire numero (numero) >** che emette il numero di bip accompagnati da lampeggi di luce.

Il blocco **< conta >**

Gli studenti generalmente non conoscono la variabile incrementale. Per affrontare questa delicata parte della programmazione, si può procedere in più fasi (vedi i successivi script nella parte della soluzione):

Fase 1: Creiamo una variabile **numero** che impostiamo a zero. Poi testiamo a turno i 5 sensori orizzontali. Se il test **“valore del sensore > 1500”**, risulta VERO aggiungiamo 1 al **numero**. Il valore del **numero** alla fine dà quindi il numero di oggetti presenti davanti ai sensori. Un primo passo concettuale importante per lo studente è notare che una variabile può cambiare il suo valore. Realizzare lo script completo per i 5 sensori e far funzionare il blocco.

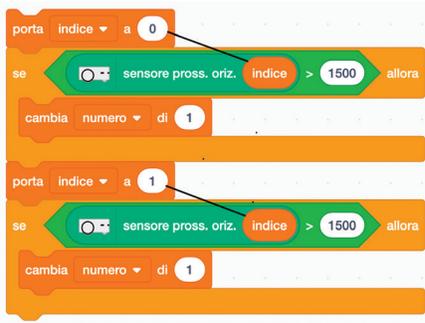


Fase 2: Dal punto di vista della programmazione, non è bene ripetere lo stesso test 5 volte con il solo cambio del riferimento al sensore.

Fortunatamente, si può sostituire il nome del sensore con un numero intero chiamato **indice** con la corrispondenza seguente:

0: sinistra; 1: primo centrale a sinistra; 2: secondo centrale a sinistra; 3: primo centrale a destra; 4: secondo centrale a destra; 5: destra

All'inizio, manteniamo i 5 test quando facciamo la sostituzione. La figura mostra un frammento di codice con 2 prove concatenate: Modificare lo script e farlo funzionare.



Fase 3: Sostituire il blocco **< impostare [indice] su (...)>** con **< aggiungere (1) a [indice]>**. Modificare lo script e farlo funzionare.

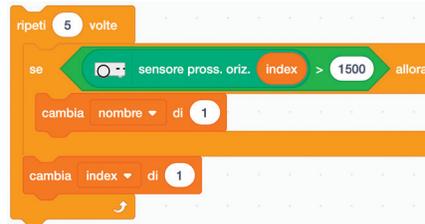
Fase 4: Leggendo lo script, possiamo vedere che ripetiamo rigorosamente 5 volte gli stessi blocchi. Da qui l'idea di utilizzare un loop di ripetizione automatico.

Il blocco **< dire il numero >**

Una volta calcolato il **numero**, viene creato un blocco che prende come numero di ingresso.

In questo blocco la sequenza viene ripetuta più volte:

- accendere i led RVB
- suonare una nota per un breve periodo
- aspettare lo stesso tempo
- spegnere i led
- aspettare un altro tempo, così i bip non si avvicinano troppo.



Il programma completo

Consiste semplicemente in:

- il blocco tipo cappello per avviare lo script.
- il blocco **< conta >**.
- il blocco **< dire il numero >**.

Estendere il programma

- Nel programma precedente, se non c'è nessun oggetto, allora **numero** = 0 quindi non c'è suono. Possiamo aggiungere in questo caso un bip diverso (un suono di sistema) e un colore diverso.

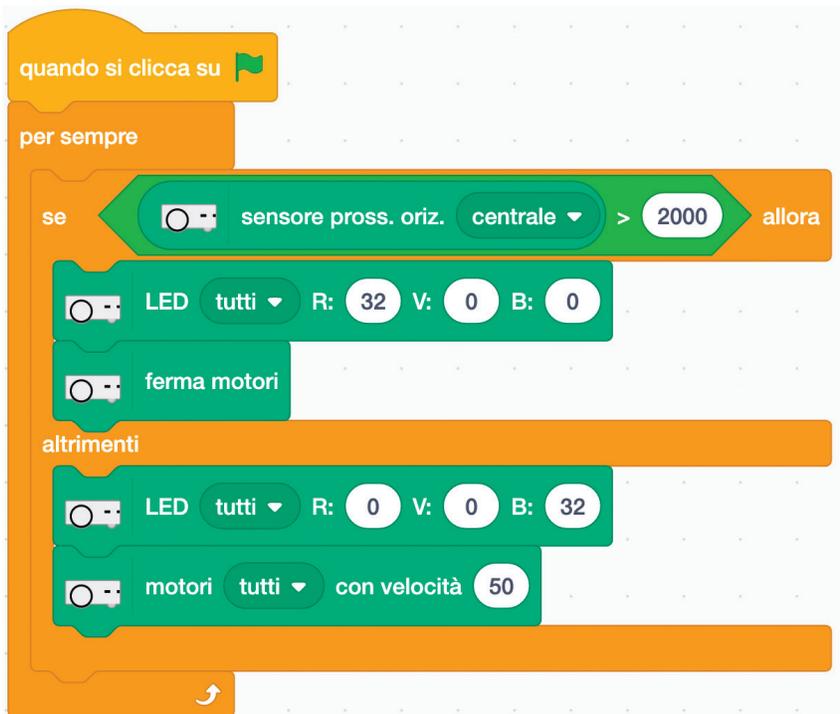
- Un'altra idea interessante (più difficile) è quella di attivare un conteggio automatico se si aggiunge o si rimuove un oggetto. Per fare questo, nel programma principale, aggiungiamo una variabile **numero_precedente** che è un altro numero all'inizio.

- Poi aggiungiamo un ciclo infinito che conta continuamente. (aggiungere un blocco **< attendere (0,5) secondi >** per evitare di bloccare Scratch). Quando aggiungiamo o rimuoviamo un oggetto, il numero diventa diverso da **numero_precedente**. Quindi mettiamo nel ciclo un test **< Se ... >** che esegue **< dire il numero >** se il numero è diverso dal **numero_precedente**. Poi si rimette il **numero_precedente** sulla variabile **numero** per evitare di eseguire di nuovo **< dire il numero >**.

Soluzioni delle attività

SOLUZIONE ATTIVITÀ S-01

Scopri i primi blocchi Scratch Thymio



SOLUZIONE ATTIVITÀ S-02

Thymio al museo

Linguaggio naturale

Se il sensore centrale non rileva nulla, allora Thymio avanzerà alla velocità 100.

Se Thymio rileva un ostacolo a sinistra, gira a destra e se rileva un ostacolo a destra, gira a sinistra.

Se il sensore centrale rileva qualcosa (c'è un oggetto proprio davanti ad esso), Thymio si muove all'indietro, girando per evitare l'ostacolo.

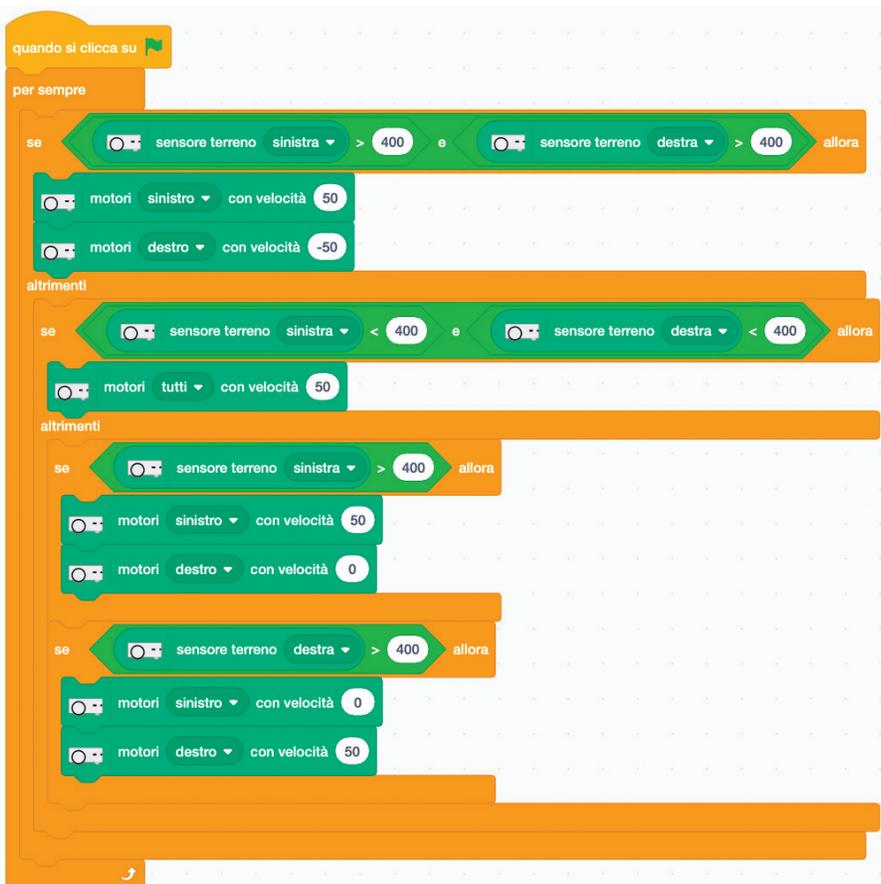
Script modificato

The image shows a Scratch script for controlling a Thymio robot. The script is organized as follows:

- quando si clicca su** (when clicked):
 - per sempre** (forever loop):
 - se** (if) block: **sensore pross. oriz. centrale** < **1000** allora (then).
 - motori tutti** con velocità **50** (set all motors to speed 50).
 - se** (if) block: **sensore pross. oriz. sinistra** > **2000** allora (then).
 - motori sinistro** con velocità **75** (set left motor to speed 75).
 - motori destro** con velocità **0** (set right motor to speed 0).
 - se** (if) block: **sensore pross. oriz. destra** > **2000** allora (then).
 - motori sinistro** con velocità **0** (set left motor to speed 0).
 - motori destro** con velocità **75** (set right motor to speed 75).
 - altrimenti** (otherwise) block:
 - ferma motori** (stop motors).

SOLUZIONE ATTIVITÀ S-03

Thymio segue la linea



The image shows a Scratch script designed to control a Thymio robot for line following. The script is organized as follows:

- Start:** A "quando si clicca su" (when clicked) event block.
- Forever Loop:** A "per sempre" (forever) loop containing:
 - Condition:** A "se" (if) block with two "e" (and) conditions: "sensore terreno sinistra > 400" and "sensore terreno destra > 400".
 - Actions:** "motori sinistro con velocità 50" and "motori destro con velocità -50".
- Else:** An "altrimenti" (else) block with:
 - Condition:** A "se" (if) block with two "e" (and) conditions: "sensore terreno sinistra < 400" and "sensore terreno destra < 400".
 - Action:** "motori tutti con velocità 50".
- Else-If:** A second "altrimenti" (else) block with:
 - Condition:** A "se" (if) block: "sensore terreno sinistra > 400".
 - Actions:** "motori sinistro con velocità 50" and "motori destro con velocità 0".
- Else-If:** A third "altrimenti" (else) block with:
 - Condition:** A "se" (if) block: "sensore terreno destra > 400".
 - Actions:** "motori sinistro con velocità 0" and "motori destro con velocità 50".

Thymio sotto controllo

Controllo con i pulsanti

Scratch script for button control:

- quando si preme il pulsante avanti
 - motori tutti con velocità 50
- quando si preme il pulsante sinistro
 - motori sinistro con velocità 15
 - motori destro con velocità 50
- quando si preme il pulsante dietro
 - motori tutti con velocità -50
- ferma motori
- quando si preme il pulsante destro
 - motori sinistro con velocità 50
 - motori destro con velocità 15

Controllo con la tastiera

Scratch script for keyboard control:

- quando si preme il tasto freccia su
 - motori tutti con velocità 50
- quando si preme il tasto freccia sinistra
 - motori sinistro con velocità 15
 - motori destro con velocità 50
- quando si preme il tasto freccia destra
 - motori sinistro con velocità 50
 - motori destro con velocità 15
- quando si preme il tasto freccia giù
 - motori tutti con velocità -50
- quando si preme il tasto spazio
 - ferma motori

SOLUZIONE ATTIVITÀ S-05

La staffetta Thymio



SOLUZIONE ATTIVITÀ S-06

Il musicista (1)

Suonare con le dita

The code is organized into five separate blocks, each triggered by a click on a specific sensor. Each block follows a similar structure: a 'when clicked' event, a 'wait until' condition, an LED configuration, a note sound, a wait time, and an LED off command.

- Block 1 (Top):** Triggered by 'quando si clicca su'. Waits until 'sensore pross. oriz. centro sinistra > 2000'. Sets LED R: 0, V: 32, B: 32. Plays note 'la' at 440 Hz for 'duree' seconds. Waits 'duree' seconds, then turns LED off.
- Block 2:** Triggered by 'quando si clicca su'. Waits until 'sensore pross. oriz. sinistra > 2000'. Sets LED R: 0, V: 0, B: 32. Plays note 'sol' at 392 Hz for 'duree' seconds. Waits 'duree' seconds, then turns LED off.
- Block 3:** Triggered by 'quando si clicca su'. Waits until 'sensore pross. oriz. centrale > 2000'. Sets LED R: 0, V: 32, B: 0. Plays note 'si' at 494 Hz for 'duree' seconds. Waits 'duree' seconds, then turns LED off.
- Block 4:** Triggered by 'quando si clicca su'. Waits until 'sensore pross. oriz. destra > 2000'. Sets LED R: 32, V: 0, B: 0. Plays note 'ré' at 587 Hz for 'duree' seconds. Waits 'duree' seconds, then turns LED off.
- Block 5 (Bottom):** Triggered by 'quando si clicca su'. Waits until 'sensore pross. oriz. centro destra > 2000'. Sets LED R: 32, V: 32, B: 0. Plays note 'do' at 523 Hz for 'duree' seconds. Waits 'duree' seconds, then turns LED off.

SOLUZIONE ATTIVITÀ S-06

Il musicista (2)

Suona una melodia

The code is organized into five columns:

- Column 1 (sol):** A pink block 'definisci sol', followed by a green block 'LED tutti R: 0 V: 0 B: 32', a green block 'suona nota a 392 Hz per durata s', an orange block 'attendi durata secondi', and a green block 'spegni LED'.
- Column 2 (la):** A pink block 'definisci la', followed by a green block 'LED tutti R: 0 V: 32 B: 32', a green block 'suona nota a 440 Hz per durata s', an orange block 'attendi durata secondi', and a green block 'spegni LED'.
- Column 3 (si):** A pink block 'definisci si', followed by a green block 'LED tutti R: 0 V: 32 B: 0', a green block 'suona nota a 494 Hz per durata s', an orange block 'attendi durata secondi', and a green block 'spegni LED'.
- Column 4 (do):** A pink block 'definisci do', followed by a green block 'LED tutti R: 32 V: 32 B: 0', a green block 'suona nota a 523 Hz per durata s', an orange block 'attendi durata secondi', and a green block 'spegni LED'.
- Column 5 (Melody):** A yellow block 'quando si clicca su', followed by an orange block 'porta durata a 0.5', and a vertical stack of notes: 'sol', 'la', 'sol', 'si', 'do', 'si', 'do', 'la', 'sol', each followed by an orange block 'attendi 0.5 secondi'.

SOS Gas tossici

The image displays four Scratch code blocks for defining functions:

- definisci S**
 - fai un cerchio di raggio 60 per -150
 - fai un cerchio di raggio 60 per 130
- definisci linea-O-linea**
 - avanza di 80
 - fai un cerchio di raggio 120 per 390
 - avanza di 240
- quando si clicca su**
 - S
 - linea-O-linea
 - gira-S
- definisci gira-S**
 - ruota di 180
 - fai un cerchio di raggio 60 per -150
 - fai un cerchio di raggio 60 per 150

Thymio sta perdendo il controllo

```

quando si clicca su
  porta colori a 0
  porta Velocità a 20
  ripeti fino a quando Velocità > 155
    motori sinistro con velocità 0 - Velocità
    motori destro con velocità Velocità
    LED tutti R: colori V: 0 B: 0
    attendi 0.2 secondi
    cambia Velocità di 5
    cambia colori di 0.5
  ripeti 4 volte
    accendere suono sistema 4
    attendi 0.5 secondi
  ripeti fino a quando Velocità = 0
    motori sinistro con velocità 0 - Velocità
    motori destro con velocità Velocità
    LED tutti R: colori V: 0 B: 0
    attendi 0.2 secondi
    cambia Velocità di -5
    cambia colori di -0.5
  attendi 3 secondi
  accendere suono sistema 5
  spegni LED
  LED temperatura R: 0 B: 0
  LED sensori terreno 0 0
  ferma motori
  
```

Dobbiamo salvare il robot Susan (1)

Scripts Asimov

The image displays six Scratch scripts arranged in a 3x2 grid, designed to control a Thymio robot. Each script is triggered by a specific event and contains several motor control blocks.

- Top Left:** Triggered by "quando si clicca su" (when clicked), it sets a "variabile" (variable) to 30 and then moves "motori tutti" (all motors) with "velocità" (speed) "variabile".
- Top Right:** Triggered by "quando si preme il tasto freccia destra" (when right arrow key is pressed), it sets "motori sinistro" (left motor) speed to "variabile" and "motori destro" (right motor) speed to 0.
- Middle Left:** Triggered by "quando si preme il tasto freccia su" (when up arrow key is pressed), it increases the "variabile" by 10 and moves "motori tutti" with speed "variabile".
- Middle Right:** Triggered by "quando si preme il tasto freccia sinistra" (when left arrow key is pressed), it sets "motori sinistro" speed to 0 and "motori destro" speed to "variabile".
- Bottom Left:** Triggered by "quando si preme il tasto freccia giù" (when down arrow key is pressed), it decreases the "variabile" by 10 and moves "motori tutti" with speed "variabile".
- Bottom Right:** Triggered by "quando si preme il tasto spazio" (when space key is pressed), it executes "ferma motori" (stop motors).

Dobbiamo salvare il robot Susan (2)

Script in linguaggio naturale

Sezione movimenti

impostare guasto a 0

aumentare velocità fino a 50

spegnere i LED

ripetere fino a guasto = 1

se la distanza (davanti) = 190 o se la distanza (sotto) > 250 allora

..... spegnere i motori

altrimenti

..... se il sensore centrale > 1500 allora

..... accendere i motori

..... altrimenti

..... se i sensori a sinistra > 2000

..... solo il motore destro gira a velocità, l'altro è fermo.

..... se i sensori di destra > 2000

..... solo il motore sinistro gira a velocità, l'altro è fermo.

Sezione dei comportamenti

mettere tempo_totale = 90 // 90 secondi per controllo. Per fare i test, mettere meno tempo

impostare guasto a 0

azzerare il cronometro

Ripetere fino a quando il cronometro non supera tempo_totale

..... Top LED verde

aspetta 1 secondo

..... LED in alto spento

aspetta 1 secondo

guasto a 1

ripetere 5 volte

..... riprodurre il sistema audio 5

LED in alto rosso

attendere 0,5 secondi

..... LED in alto spento

attendere 0,5 secondi

spegnere i LED

spegnere i motori

riprodurre il sistema audio 1

Dobbiamo salvare il robot Susan (2)

Scripts Susan

Script 1: quando si clicca su [bandierina verde]

- porta Velocità a 100
- LED superiori R: 0 V: 0 B: 0
- ripeti fino a quando guasto = 1
- se distanza davanti = 190 o distanza terreno > 250 allora
 - ferma motori
- altrimenti
 - se sensore pross. oriz. centrale > 1500 allora
 - motori tutti con velocità Velocità
 - altrimenti
 - se sensore pross. oriz. sinistra > 2000 allora
 - motori sinistro con velocità 0
 - motori destro con velocità Velocità
 - se sensore pross. oriz. destra > 2000 allora
 - motori sinistro con velocità Velocità
 - motori destro con velocità 0
- attendi 0.2 secondi

Script 2: quando si clicca su [pulsante]

- porta tempo_totale a 10
- porta guasto a 0
- azzerà cronometro
- ripeti fino a quando cronometro > tempo_totale
 - LED superiori R: 0 V: 32 B: 0
 - attendi 1 secondi
 - LED superiori R: 0 V: 0 B: 0
 - attendi 1 secondi
- porta guasto a 1
- ripeti 5 volte
 - accendere suono sistema 5
 - LED tutti R: 32 V: 0 B: 0
 - attendi 0.5 secondi
 - LED tutti R: 0 V: 0 B: 0
 - attendi 0.5 secondi
- spigni LED
- ferma motori
- accendere suono sistema 1

SOLUZIONE ATTIVITÀ S-II

Il gioco della bandiera (1)

Soluzione senza variabili

The script is composed of several event-driven blocks:

- quando si clicca su** (when clicked)
 - per sempre** (forever loop)
 - se** (if) condition: `sensore pross. oriz. posteriore sinistro > 2500` o `sensore pross. oriz. posteriore destro > 2500` allora
 - ripeti 3 volte** (repeat 3 times)
 - LED tutti** (LED all) with values: R: 32 V: 0 B: 0
 - attendi 0.5 secondi** (wait 0.5 seconds)
 - LED tutti** (LED all) with values: R: 0 V: 0 B: 0
 - attendi 0.5 secondi** (wait 0.5 seconds)
- quando si preme il tasto freccia sinistra** (when key pressed: left arrow)
 - motori sinistro** (motors left) con velocità 15 (with speed 15)
 - motori destro** (motors right) con velocità 50 (with speed 50)
- quando si preme il tasto freccia su** (when key pressed: up arrow)
 - motori tutti** (motors all) con velocità 50 (with speed 50)
- quando si preme il tasto freccia destra** (when key pressed: right arrow)
 - motori sinistro** (motors left) con velocità 50 (with speed 50)
 - motori destro** (motors right) con velocità 15 (with speed 15)
- quando si preme il tasto spazio** (when key pressed: space)
 - ferma motori** (stop motors)
- quando si preme il tasto freccia giù** (when key pressed: down arrow)
 - motori tutti** (motors all) con velocità -50 (with speed -50)

Il gioco della bandiera (2)

Soluzione a velocità progressiva

quando si clicca su

porta velocità a 0

per sempre

se $\text{sensore pross. oriz. posteriore sinistro} > 2500$ o $\text{sensore pross. oriz. posteriore destro} > 2500$ allora

ripeti 3 volte

LED tutti R: 32 V: 0 B: 0

attendi 0.5 secondi

LED tutti R: 0 V: 0 B: 0

attendi 0.5 secondi

quando si preme il tasto freccia sinistra

motori sinistro con velocità 0

motori destro con velocità velocità

quando si preme il tasto freccia su

cambia velocità di 5

motori tutti con velocità velocità

motori destro con velocità 0

quando si preme il tasto freccia destra

motori sinistro con velocità velocità

motori destro con velocità 0

quando si preme il tasto spazio

ferma motori

quando si preme il tasto freccia giù

cambia velocità di -5

motori tutti con velocità velocità

Il labirinto

```
quando si clicca su [bandierina]
  ripeti fino a quando [tasto spazio premuto]
    se [sensore pross. oriz. centrale > 1500 o sensore pross. oriz. sinistra > 1500] allora
      motori sinistro con velocità 40
      motori destro con velocità 10
    altrimenti
      motori sinistro con velocità 10
      motori destro con velocità 40
  ferma motori
```

The image shows a Scratch script designed for a Thymio robot. The script starts with a 'when green flag clicked' event. It then enters a 'repeat until' loop that continues as long as the 'space' key is pressed. Inside this loop, there is an 'if-else' conditional block. The 'if' condition checks if either the central proximity sensor is greater than 1500 or the left proximity sensor is greater than 1500. If true, it sets the left motor to 40% speed and the right motor to 10% speed. If false, it sets the left motor to 10% speed and the right motor to 40% speed. After the loop, there is a 'stop motors' block.

Thymio sul banco di prova

```

quando si clicca su [bandierina]
  porta altezza a 100
  porta velocita a 5
  ripeti fino a quando [velocita > 150]
    motori tutti con velocita velocita
    cambia velocita di 5
    porta altezza a [altezza * 1.1]
    suona nota a [altezza] Hz per 0.1 s
    attendi 0.1 secondi
  ripeti fino a quando [velocita < 0]
    motori tutti con velocita velocita
    cambia velocita di -5
    porta altezza a [altezza / 1.1]
    suona nota a [altezza] Hz per 0.1 s
    attendi 0.1 secondi
  ferma motori
  
```

```

quando si clicca su [bandierina]
  porta velocita a 5
  ripeti fino a quando [velocita > 150]
    motori tutti con velocita velocita
    cambia velocita di 5
    attendi 0.1 secondi
  ferma motori
  
```

Thymio scout

The image shows a Scratch script designed to control a Thymio robot. The script is built on a grid background and consists of the following blocks:

- quando si clicca su** (when clicked) block.
- spegni LED** (turn off LED) block.
- ripeti fino a quando** (repeat until) loop with two conditions:
 - distanza terreno > 250** (ground distance > 250)
 - o** (or)
 - sensore pross. oriz. centrale > 3000** (central horizontal proximity sensor > 3000)
- se** (if) block with condition:
 - sensore pross. oriz. sinistra > sensore pross. oriz. destra** (left horizontal proximity sensor > right horizontal proximity sensor)
 - allora** (then) block containing:
 - motori sinistro con velocità 60** (left motor with speed 60)
 - motori destro con velocità 40** (right motor with speed 40)
- altrimenti** (otherwise) block containing:
 - motori sinistro con velocità 40** (left motor with speed 40)
 - motori destro con velocità 60** (right motor with speed 60)
- se** (if) block with condition:
 - sensore pross. oriz. centrale > 3000** (central horizontal proximity sensor > 3000)
 - allora** (then) block containing:
 - LED tutti R: 32 V: 0 B: 0** (all LEDs: R=32, V=0, B=0)
 - accendere suono sistema 4** (turn on system sound 4)
- altrimenti** (otherwise) block containing:
 - LED tutti R: 0 V: 32 B: 0** (all LEDs: R=0, V=32, B=0)
- per sempre** (forever) loop with condition:
 - se** (if) block with condition:
 - sensore pross. oriz. sinistra > sensore pross. oriz. destra** (left horizontal proximity sensor > right horizontal proximity sensor)
 - allora** (then) block containing:
 - motori sinistro con velocità -40** (left motor with speed -40)
 - motori destro con velocità -60** (right motor with speed -60)
 - altrimenti** (otherwise) block containing:
 - motori sinistro con velocità -60** (left motor with speed -60)
 - motori destro con velocità -40** (right motor with speed -40)

Thymio scientifico (1)

Da dare agli studenti

The image shows a Scratch script designed for the Thymio robot. The script is composed of several blocks:

- quando si clicca su** (when clicked) block with a green flag icon.
- attendi fino a quando** (wait until) block with the condition `tasto spazio > premuto`.
- ferma motori** (stop motors) block.
- dire** (say) block with the text `risultato`.
- pensa** (think) block with the text `Segna la nuova posizione e misura la distanza percorsa in cm, quindi premi la barra spaziatrice sulla tastiera`.
- porta motore** (set motor) block with `a 30`.
- porta velocita** (set speed) block with `a distanza / fine`.
- porta risultato** (set result) block with `a unione di unione di la velocità è: e velocita e cm/s`.
- azzerà cronometro** (reset stopwatch) block.
- porta fine** (set end) block with `a cronometro`.
- pensa** (think) block with the text `Segna con una linea la posizione di Thymio e poi premi la barra spaziatrice della tastiera`.
- porta velocita** (set speed) block with `a arrotonda velocita * 100 / 100`.
- attendi fino a quando** (wait until) block with the condition `cronometro > 10`.
- porta distanza** (set distance) block with `a risposta`.
- chiedi** (ask) block with the text `Scrivi la distanza percorsa e attendi`.
- motori tutti** (all motors) block with `con velocità motore`.

SOLUZIONE ATTIVITÀ S-15

Thymio scientifico (2)

Soluzione

```

quando si clicca su
  pensa Segna con una linea la posizione iniziale di Thymio e poi premi la barra spaziatrice della tastiera
  attendi fino a quando [tasto spazio premuto]
  azzerà cronometro
  motori tutti con velocità 25
  attendi fino a quando [cronometro > 8]
  ferma motori
  porta fine a cronometro
  pensa Segna la nuova posizione e misura la distanza percorsa in cm, quindi premi la barra spaziatrice sulla tastiera.
  chiedi [Scrivi la distanza percorsa] e attendi
  porta distanza a risposta
  porta velocità a [distanza / fine]
  porta velocità a [arrotonda velocità * 100 / 100]
  porta risultato a [unione di unione di la velocità è : e velocità e cm/s]
  dire risultato
  
```

Numero esperienza	1	2	3	4	5	6	7	8
Velocità-blocco	25	50	75	100	125	150	175	200
Velocità-sperimentale	2,75	5,11	7,48	10	11,3	11,2	11,1	11,2

Thymio equilibrista



```
quando si clicca su [bandierina]
  spegni LED
  porta velocita a 0
  porta k a 10
  penna giù
  per sempre
    fai 5 passi
      porta alfa a [inclinazione davanti-dietro]
      porta velocita a [k * alfa]
      motori tutti con velocità velocita
      cambia y di velocita
      attendi 0.2 secondi
```

The image shows a Scratch script for controlling a Thymio robot. The script starts with a 'when green flag clicked' event block. It then performs a sequence of actions: turning off the LED, setting the 'velocita' port to 0, setting the 'k' port to 10, and moving the pen down. A 'per sempre' (forever) loop follows, containing five 'fai passi' (do steps) blocks. Each step block includes: setting the 'alfa' port to the 'inclinazione davanti-dietro' sensor value, setting the 'velocita' port to the product of 'k' and 'alfa', setting all motors to run at the 'velocita' port's value, changing the y-coordinate by the 'velocita' value, and waiting for 0.2 seconds before repeating the loop.

SOLUZIONE ATTIVITÀ S-17

Thymio ambulanza

Scripts per l'ambulanza

quando si clicca su

per sempre

motori tutti con velocità 150

attendi fino a quando sensore pross. oriz. centrale > 2500

ferma motori

attendi fino a quando sensore pross. oriz. centrale < 1500

quando si clicca su

per sempre

suona nota a 294 Hz per 1 s

attendi 1 secondi

suona nota a 440 Hz per 1 s

attendi 1 secondi

Script per il taxi

quando si clicca su

per sempre

attendi fino a quando livello sonoro > 10

fai un cerchio di raggio 250 per 60

fai un cerchio di raggio 250 per -60

Thymio al parcheggio

The image shows a Scratch script designed to control a Thymio robot for a parking task. The script is organized into several functional blocks:

- Initial Setup:** A 'quando si clicca su' (when clicked) event block triggers the start of the sequence. It sets the 'velocità' (velocity) to 50 and calls the 'entre_mur' (enter wall) function.
- Entrance:** The 'entre_mur' function block contains:
 - Turning the left wheel (-90) with a velocity of 50.
 - A 'recluer' (retract) block.
 - An 'attendi fino a quando' (wait until) block for the 'tasto spazio' (space key) to be pressed.
 - Advancing 300 units with a velocity of 50.
 - Turning the right wheel (120) with a velocity of 50.
 - Calling the 'entre_mur' function again.
- Defining 'entre_mur':** A 'definisci' (define) block for the 'entre_mur' function:
 - Repeats until the 'tasto spazio' (space key) is pressed.
 - Uses an 'if' block to check 'distanza davanti' (distance ahead) > 100. If true, it sets all motors to 'vitesse' (speed).
 - Otherwise, it uses a complex 'if' block with four sensors (left, center-left, center-right, right) to determine the robot's position and sets motor velocities accordingly:
 - Left sensor: left motor +vitesse, right motor -20.
 - Center-left sensor: left motor +vitesse, right motor +20.
 - Center-right sensor: left motor -20, right motor +vitesse.
 - Right sensor: left motor -20, right motor +vitesse.
- Defining 'recluer':** A 'definisci' (define) block for the 'recluer' function:
 - Sets all motors to a velocity of -50.
 - Waits until 'distanza dietro' (distance behind) < 125.
 - Repeats until 'distanza dietro' < 5.
 - Uses an 'if' block to check 'sensore pross. oriz. posteriore destro' (right rear proximity sensor) > 'sensore pross. oriz. posteriore sinistro' (left rear proximity sensor). If true, it sets left motor to -40 and right motor to -20.
 - Otherwise, it sets left motor to -20 and right motor to -40.
 - Finally, it stops the motors.

Avvio cifrato

The image shows a Scratch script for controlling a Thymio robot. The script starts with a 'quando si clicca su' (when clicked) event block. It then sets the LED 'tutti' (all) to 'R: 0 V: 0 B: 32'. The main logic consists of a series of conditional blocks: 'attendi fino a quando' (wait until) blocks with four options: 'pulsante avanti' (forward button), 'pulsante dietro' (back button), 'pulsante sinistro' (left button), and 'pulsante destro' (right button). Each 'attendi' block is followed by an 'allora' (then) block containing a 'se' (if) block with the same button option, which leads to another 'attendi 0.5 secondi' (wait 0.5 seconds) block. This sequence repeats for the forward, back, left, and right buttons. After the right button sequence, the script sets the motor 'tutti' (all) to 'con velocità 50' (with speed 50). This is followed by a series of 'altrimenti' (otherwise) blocks, each containing a 'fine' (end) block. A separate block on the right shows a 'definisci fine' (define end) block, followed by 'accendere suono sistema: 1' (turn on system sound: 1) and 'LED tutti: R: 32 V: 0 B: 0' (set all LEDs to R: 32 V: 0 B: 0).

Thymio impara a contare

Blocco contare - passo 1

```

definisci conta
porta numero a 0
se [sensore pross. orz. sinistra > 1500] allora
  cambia numero di 1
se [sensore pross. orz. centro sinistra > 1500] allora
  cambia numero di 1
se [sensore pross. orz. centrale > 1500] allora
  cambia numero di 1
se [sensore pross. orz. centro destra > 1500] allora
  cambia numero di 1
se [sensore pross. orz. destra > 1500] allora
  cambia numero di 1
  
```

Blocco contare - passo 2

```

definisci conta
porta numero a 0
porta indice a 0
se [sensore pross. orz. indice > 1500] allora
  cambia numero di 1
  porta indice a 1
se [sensore pross. orz. indice > 1500] allora
  cambia numero di 1
  porta indice a 2
se [sensore pross. orz. indice > 1500] allora
  cambia numero di 1
  porta indice a 3
se [sensore pross. orz. indice > 1500] allora
  cambia numero di 1
  porta indice a 4
se [sensore pross. orz. indice > 1500] allora
  cambia numero di 1
  
```

Blocco contare - passo 3

```

definisci conta
porta numero a 0
porta indice a 0
se [sensore pross. orz. indice > 1500] allora
  cambia numero di 1
  cambia indice di 1
se [sensore pross. orz. indice > 1500] allora
  cambia numero di 1
  cambia indice di 1
se [sensore pross. orz. indice > 1500] allora
  cambia numero di 1
  cambia indice di 1
se [sensore pross. orz. indice > 1500] allora
  cambia numero di 1
  cambia indice di 1
se [sensore pross. orz. indice > 1500] allora
  cambia numero di 1
  cambia indice di 1
se [sensore pross. orz. indice > 1500] allora
  cambia numero di 1
  
```

Blocco contare - passo 4

```

definisci conta
porta indice a 0
porta numero a 0
ripeti 5 volte
  se [sensore pross. orz. indice > seuil] allora
    cambia numero di 1
  cambia indice di 1
  
```

Thymio impara a contare

2 - Programma completo

```

quando si clicca su [bandierina]
  conta
  dire le nombre [numero]
  
```

```

definisci [conta]
  porta [numero] a [0]
  porta [indice] a [0]
  ripeti [5] volte
    se [sensore pross. oriz. indice > 1500] allora
      cambia [numero] di [1]
      cambia [indice] di [1]
  
```

```

definisci [dire il numero numero]
  
```

3 - Programma arricchito

```

quando si clicca su [bandierina]
  porta [numero] a [0]
  porta [numero precedente] a [0]
  per sempre
    comper
    se [non numero = numero precedente] allora
      dire le nombre [numero]
      porta [numero precedente] a [numero]
    attendi [0.2] secondi
  
```

Consigli per l'insegnante

MURI E LINEE NERE

Realizzare i muri

In carta o cartone sottile

Piegare un foglio A4 3 volte nel senso della lunghezza. Il risultato è un tubo a sezione quadrata. Appiattirlo e incollare i 2 lati opposti con un pezzo di adesivo, poi ridargli il suo volume. La dimensione è quindi di 297 x 52 x 52 mm. L'esperienza dimostra che queste pareti sono facili da immagazzinare, trasportare e riutilizzare.

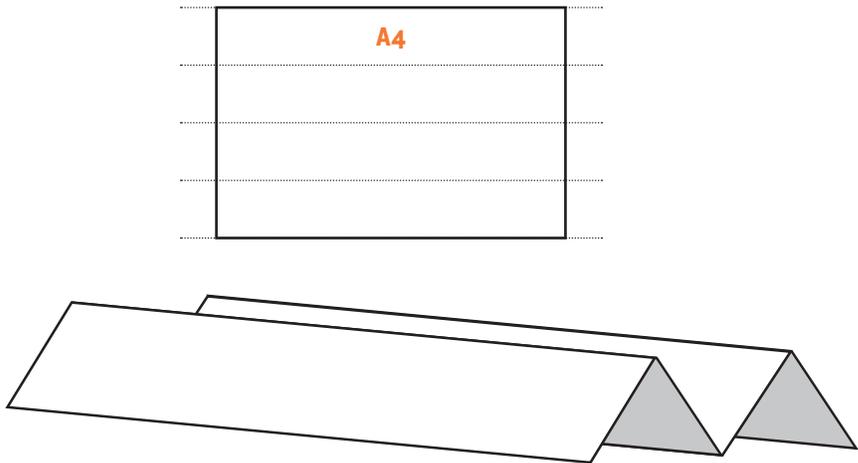
Perché non utilizzare carta già utilizzata? O far disegnare agli studenti dei cartoncini?

In legno di balsa

Tagliare le strisce in sezioni di 30 cm, con un'altezza minima di 5 cm per assicurarsi che i raggi ad infrarosso dei sensori di prossimità possano essere riflessi dalle pareti.

Realizzare linee nere

Le linee devono avere una larghezza minima di 4 cm perchè possano essere rilevate dai sensori del terreno sotto al robot. Anche le tempere possono essere utilizzate per tracciare le linee ma hanno il problema di bagnare la carta e deformarla. Per un risultato più efficace, è possibile utilizzare nastro nero largo 5 cm (nastro isolante da elettricista).



Consigli per l'insegnante

NOTE E COLORI

Ottava	1	2	3	4
Do	65 Hz	131 Hz	262 Hz	523 Hz
Do diesis o re bemolle	69 Hz	139 Hz	277 Hz	554 Hz
Re	74 Hz	147 Hz	294 Hz	587 Hz
Re diesis o mi bemolle	78 Hz	156 Hz	311 Hz	622 Hz
Mi	83 Hz	165 Hz	330 Hz	659 Hz
Fa	87 Hz	175 Hz	349 Hz	698,5 Hz
Fa diesis o sol bemolle	92,5 Hz	185 Hz	370 Hz	740 Hz
Sol	98 Hz	196 Hz	392 Hz	784 Hz
Sol diesis o La bemolle	104 Hz	208 Hz	415 Hz	831 Hz
La	110 Hz	220 Hz	440 Hz	880 Hz
La diesis o Si bemolle	117 Hz	233 Hz	466 Hz	932 Hz
Si	123 Hz	247 Hz	494 Hz	988 Hz

Codici colore

colore	rosso	verde	blu	giallo	ciano	magenta
R	32	0	0	32	0	32
V	0	32	0	32	32	0
B	0	0	32	0	32	32

Blocchi Scratch per Thymio

Indice

- 1. Informazioni generali
 - 2. Blocchi di movimento
 - 3. Blocchi di LED luminosi
 - 4. Blocchi di suono
 - 5. Blocchi eventi
 - 6. Blocchi logici
 - 7. Blocchi sensore
-

1. Informazioni generali

In Thymio Suite, dopo aver lanciato Scratch, troviamo i blocchi specifici del robot nel menu a icone verticali a sinistra, in basso.

Ci sono 45 blocchi Scratch Thymio in questa versione, ma è probabile che questo numero si evolva in futuro. Possiamo classificare questi blocchi in 4 categorie facilmente identificabili per la loro forma:

Blocchi attuatori



Riconoscibili grazie alle loro anse che permettono di impilarli.

Trasmettono un comando al robot per compiere un'azione che quest'ultimo deve eseguire.

Gli attuatori sono di tre tipi:

- di movimento
- LED luminosi
- di suono (attraverso un altoparlante).

Blocchi sensore



Riconoscibili grazie alle loro estremità arrotondate che permettono di inserirli in altri blocchi.

Questi blocchi restituiscono le misure effettuate dal robot. Si trovano queste tipologie:

- distanza, grazie ai sensori trasmettitori/ricevitori a infrarossi
- suono, attraverso un microfono
- shock/inclinazione, grazie ad un accelerometro
- velocità misurata del motore.

Blocchi sensori logici



Riconoscibile per le punte affilate.

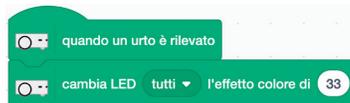
Restituisce un'espressione booleana, quindi vera o falsa.

Blocchi evento

Sono sempre in cima a uno script e sono attivi non appena compaiono nell'area corrispondente.



Ad esempio, la sequenza qui indicata accenderà il robot in rosso non appena si clicca su di essa, anche se la bandierina verde di Scratch non è stata premuta.



Convenzione tipografica adottata per i blocchi:

Un blocco è definito tra i 2 segni minore < e maggiore >.

Le parentesi quadre indicano la presenza di un elenco di opzioni, separate dal simbolo |.

Le parentesi tonde indicano un cerchio dove inserire un blocco sensore, una variabile o un numero intero relativo.

I valori predefiniti sono indicati in grassetto.

Il termine numero si riferisce ad un numero decimale. Il separatore decimale è il punto.

Il termine intero si riferisce ad un numero intero.

2. Blocchi di movimento

< motori [opzione] con velocità (velocità) >



Attiva la rotazione del motore o dei motori in modo che la ruota o le ruote si muovano in avanti.

- **opzione:** sinistra | destra | tutto : corrisponde alla sinistra, alla destra o ad entrambe le ruote (Thymio è visto da dietro)
- **valore predefinito:** tutti
- **velocità:** velocità di avanzamento di Thymio
- **valore predefinito:** 50
- **limiti:** da -150 a 150.

Nota 1: L'unità di misura più adatta alla velocità è vicina a mm/s. Ma questo può variare a seconda della natura del terreno su cui si muove Thymio.

Nota 2: Nel caso dell'opzione **tutti**, la traiettoria del robot non è molto diritta, specialmente se la velocità è elevata. Pertanto, è necessario adottare velocità modeste se non si vuole che Thymio devii troppo a sinistra o a destra. È comunque possibile migliorare la traiettoria calibrando i motori.

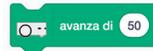
< ferma motori >



Fa sì che le ruote si arrestino.

Nota preliminare: per tutti i blocchi del tipo di avanzamento o rotazione, durante il movimento, Thymio è "sordo" a tutti gli eventi o cambiamenti di valori provenienti dai sensori.

< avanza di (distanza) >



Thymio va dritto, poi si ferma.

- **distanza:** distanza percorsa da Thymio. L'unità corrisponde al mm, ma è abbastanza approssimativa
- **valore di default:** 50
- **limitazioni:** la principale è la tendenza a girare sempre di più con l'aumentare della distanza. È necessario effettuare alcuni test. Il Thymio si muove all'indietro con valori negativi.

Nota 1: durante il movimento, Thymio non percepisce alcun evento.

Nota 2: per distanza = 0, Thymio avanza senza fermarsi.

< avanza di (distanza) con (velocità) >



Thymio va dritto, poi si ferma.

Distanza: distanza percorsa da Thymio. Unità: mm (approssimativo)

- **valore di default:** 50
- **velocità:** la velocità di avanzamento di Thymio. Unità: mm/s (approssimativo)
- **valore predefinito:** 50
- **limiti:** da -150 a 150.

< avanza di (distanza) in (tempo) s >



Thymio va dritto, poi si ferma.

- **distanza:** distanza percorsa da Thymio. Unità: mm (approssimativo)
- **valore di default:** 50
- **tempo:** tempo di viaggio. Unità: secondi (approssimativo)
- **valore predefinito:** 1
- **limitazioni:** non accetta valori inferiori a 1. Il limite principale è la tendenza a curvare sempre di più con l'aumentare del tempo di viaggio. È necessario effettuare dei test.

< ruota di (angolo) >

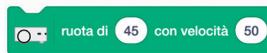


Thymio si gira sul posto, poi si ferma. La velocità di rotazione è di 40 (vedere il prossimo blocco).

- **angolo:** angolo di rotazione in gradi (approssimativo)
- se l'angolo è positivo, gira a destra, se l'angolo è negativo, gira a sinistra
- **valore di default:** 45
- **limitazioni:** il limite principale è l'errore sull'angolo che aumenta con il suo valore. Va notato tuttavia che l'errore tra l'angolo e il valore reale è riproducibile, per cui possiamo cercare di effettuare correzioni efficaci.

Il centro di rotazione è il centro del robot, con le ruote che vanno in direzioni opposte alla stessa velocità.

< ruota di (angolo) con velocità (velocità) >



Thymio si gira sul posto, poi si ferma.

- **angolo:** angolo di rotazione in gradi (approssimativo)
- se l'angolo è positivo, gira a destra, se l'angolo è negativo, gira a sinistra
- **valore di default:** 90
- **velocità:** velocità di rotazione di Thymio. Unità arbitraria: 100 corrisponde a 125 gradi/s
- **valore predefinito:** 50
- **limitazioni:** la limitazione principale è l'errore sull'angolo, che aumenta con la velocità. Fino a 100 gradi, il movimento angolare è rispettato.

< ruota di (angolo) in (tempo) s >



Thymio si gira sul posto, poi si ferma.

- **angolo:** angolo di rotazione in gradi (approssimativo)
- **valore di default:** 90
- **tempo:** tempo di rotazione. Unità: secondi (approssimativo)
- **valore predefinito:** 1
- **limitazioni:** non accetta valori inferiori a 1. Il problema principale è la velocità della ruota. Ad esempio, la rotazione di 360° in 1 secondo provoca una rotazione solo di circa 180°.

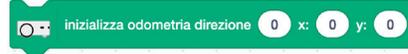
< fai un cerchio di raggio (distanza) per (angolo) >



Thymio avanza girando su un percorso circolare e poi si ferma.

- **distanza:** distanza percorsa sull'arco di cerchio sotteso dall'angolo (approssimativo)
- **valore di default:** 150
- **limitazioni:** accetta valori negativi
- **angolo:** angolo di rotazione in gradi (approssimativo)
- **valore di default:** 45.

< **inizializza odometria direzione**
(direzione) x: (posizione) y: (posizione)>



La posizione del robot sarà definita come la posizione iniziale rispetto ai valori forniti dal sensore del contachilometri.

Per la compatibilità con Scratch, la direzione iniziale è impostata a 90°.

- **angolo:** angolo in gradi, orientato come i blocchi <ruota>, positivo per una rotazione a destra
- **valore di default:** 90
- **x, y:** proiezione ortogonale del vettore di posizione Thymio, unità identica a quella dei blocchi <avanza>
- **valore predefinito:** 0,0.

3. Blocchi LED luminosi

< **LED [opzione] R: (intensitàR) V: (intensitàV) B:**
(intensitàB) >



Accendere i LED RVB Thymio.

- **opzioni:** tutte | in alto | in basso | a sinistra in basso | a destra in basso

Thymio ha 2 LED RVB multicolore in alto che funzionano insieme, poi 2 LED sotto a destra e a sinistra che funzionano in modo indipendente. L'opzione tutti illumina tutti e 4 i LED.

- **valore predefinito:** tutti
- **intensity R:** controlla l'intensità della componente rossa
- **valore predefinito:** 32
- **limiti:** da 0 (nessuna luce) a 32 (intensità massima)
- **intensità V:** controlla l'intensità della componente verde
- **valore predefinito:** 0
- **limiti:** da 0 (nessuna luce) a 32 (intensità massima)
- **intensità B:** controlla l'intensità della componente blu
- **valore predefinito:** 32
- **limiti:** da 0 (nessuna luce) a 32 (intensità massima).

< **porta LED [opzione] effetto colore**
a (tinta) >



Accendere i LED RVB Thymio.

- **opzioni:** tutte | in alto | in basso | a sinistra in basso | a destra in basso

Thymio ha 2 LED RVB multicolore in alto che funzionano insieme, poi 2 LED sotto a destra e

a sinistra che funzionano in modo indipendente. L'opzione tutti illumina tutti e 4 i LED.

• **Colore:** definisce una corrispondenza valore – colore. Quando il valore va da 0 a 198, il colore passa attraverso lo spettro dell'arcobaleno secondo la seguente scala:

Tinta	0	20	40	60	80	100	120	140	160	180
Colore	rosso	giallo arancione	giallo verde	verde	ciano	blu chiaro	blu	blu viola	magenta	fucsia

Se il valore è > 198, vengono utilizzati nuovamente gli stessi colori. Quindi $198 + 20 = 218$ restituisce lo stesso colore di 20.

• **valore predefinito:** 0

• **limitazioni:** il valore può essere molto grande ma rispetta la regola del modulo 198.

Per i valori negativi, il colore è sempre giallo.

< **cambia LED [opzione] l'effetto colore di (offset)** > _____



Cambia il colore dei LED RVB di Thymio.

• **opzioni:** tutte | in alto | in basso | a sinistra in basso | a destra in basso

Thymio ha 2 LED RVB multicolore in alto che funzionano insieme, poi 2 LED sotto a destra e a sinistra che funzionano in modo indipendente. L'opzione tutti illumina tutti e 4 i LED.

• **offset:** viene aggiunto al valore che dà il colore dei LED secondo il blocco < Mettere l'effetto colore del LED >

• **valore predefinito:** 33; permette, con aggiunte successive, di ottenere colori che non si riproducono in modo identico

• **limitazioni:** il valore può essere molto grande ma rispetta la regola del modulo 198.

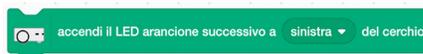
Per i valori negativi, il colore è sempre giallo.

< **Spegni LED** > _____



• Spegne tutti i LED tranne i LED associati ai sensori.

< **accendi il LED darancione successivo [opzione] del cerchio** > _____



• Se nessuno dei LED gialli nel cerchio superiore di Thymio è acceso, accendere quello davanti al pulsante anteriore

• in caso contrario, spegnere il LED acceso e accendere quello successivo, girando in una direzione che dipende dal parametro dell'opzione

• **opzione:** sinistra | destra. Sinistra: la direzione è in senso orario; destra, invece, la direzione è come in trigonometria (ovvero l'altra direzione).

< **LED cerchio (int) (int) (int) (int) (int) (int) (int) (int)** > _____



Uno ad uno, accende i LED gialli del cerchio intorno ai pulsanti.

Il primo LED si trova davanti. I seguenti sono posizionati in senso orario.

• **Int:** intensità della luce

- valori predefiniti: 0 8 16 32 0 8 8 16 32
- limiti: 0 - 32
- accetta numeri o variabili

< LED sensori pross. e orizz.
(int) (int) (int) (int) (int)
(int) (int) (int) >



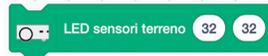
Accende i LED accanto ai sensori orizzontali.

Il primo numero controlla il LED anteriore sinistro, e così via fino al quinto che controlla il LED anteriore destro, il sesto il LED posteriore sinistro e il settimo il LED posteriore destro.

- int: intensità della luce
- valori predefiniti: 0 8 16 32 0 8 8 16 32
- limiti: 0 - 32
- accetta numeri o variabili
- l'uso di questo blocco disabilita il comportamento predefinito di questi LED, cioè di produrre un'intensità luminosa proporzionale alla radiazione catturata. Tuttavia, il sensore stesso continua a funzionare normalmente.

Per ripristinare il normale comportamento di questi LED, Scratch deve essere chiuso e poi riaperto.

< LED sensori terreno (int) (int) >



Accende i LED posti davanti, sotto Thymio, accanto ai sensori terreno.

Il primo numero controlla il LED sinistro, il secondo il LED destro.

- int : intensità di luce
- valori predefiniti: 32 - 32
- limiti: 32 - 32
- accetta numeri o variabili
- L'uso di questo blocco disabilita il comportamento predefinito di questi LED, cioè di produrre un'intensità luminosa proporzionale alla radiazione catturata. Tuttavia, il sensore stesso continua a funzionare normalmente.

Per ripristinare il normale comportamento di questi LED, Scratch deve essere chiuso e poi riaperto.

< LED pulsanti (int) (int) (int) (int) >



Accende i LED accanto ai 4 tasti freccia.

Il primo numero controlla il LED anteriore, il secondo il LED destro, il terzo il LED posteriore e il quarto il LED sinistro.

- Int: intensità della luce
- valori predefiniti: 16 32 16 32
- limiti: 32 - 32
- accetta numeri o variabili
- l'uso di questo blocco disabilita il comportamento predefinito di questi LED, cioè di produrre un'intensità luminosa proporzionale alla vicinanza di un dito. Tuttavia, il pulsante stesso continua a funzionare normalmente.

Per ripristinare il normale comportamento di questi LED, è necessario chiudere e poi riaprire Scratch.

< LED temperatura R (int) B (int) >



Accende i LED sul lato sinistro di Thymio, vicino alla ruota.
Il primo numero controlla il LED rosso, il secondo il LED blu.

- **int: intensità della luce**
- **valori predefiniti: 32 - 8**
- **limiti: 32 - 32**
- **accetta numeri o variabili**
- **l'uso di questo blocco disabilita il comportamento predefinito di questi LED, cioè di produrre un'intensità luminosa proporzionale alla temperatura del sensore. Tuttavia, il pulsante stesso continua a funzionare normalmente.**

Per ripristinare il normale comportamento di questi LED, Scratch deve essere chiuso e poi riaperto.

< LED telecomando >



Accende il LED rosso sul lato destro di Thymio, vicino alla ruota.
Il numero controlla il LED.

- **int: intensità della luce**
- **valore predefinito: 16**
- **limiti: 32 - 32**
- **accetta numeri o variabili**
- **l'uso di questo blocco disabilita il comportamento predefinito di questi LED, cioè di produrre un'intensità luminosa proporzionale al segnale del telecomando.**

Per ripristinare il normale comportamento di questi LED, è necessario chiudere e poi riaprire Scratch.

< LED microfono >



Accende il LED blu sul lato destro di Thymio, vicino alla ruota.
Il numero controlla il LED.

- **int: intensità della luce**
- **valore predefinito: 32**
- **limiti: 32 - 32**
- **accetta numeri o variabili**
- **l'uso di questo blocco disabilita il comportamento predefinito di questi LED, cioè di produrre un'intensità luminosa proporzionale al livello sonoro.**

Per ripristinare il normale comportamento di questi LED, è necessario chiudere e poi riaprire Scratch.

4. Blocchi di suono

< Accendere suono di sistema [opzione] >



Riproduci un gruppo di note predefinite.

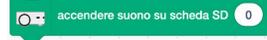
- **opzione:** da 0 a 7
- **-1:** smettere di emettere il suono
- **suono 0:** suono di avvio
- **suono 1:** suono di chiusura
- **suono 2:** suono dei pulsanti
- **suono 3:** suono del pulsante centrale
- **suono 4:** suono di stress (pauroso)
- **suono 5:** suono d'urto
- **suono 6:** suono di tracciamento dell'oggetto
- **suono 7:** suono di rilevamento oggetti per il tracciamento
- **valore predefinito:** 1.

< suonare nota a (frequenza) Hz per (durata) s >



- Suona una nota musicale con un suono sintetizzato.
- **frequenza:** le frequenze dei suoni emessi possono corrispondere alle note musicali senza comunque essere molto precise. L'esempio 440 corrisponde alla nota **La** e 294 alla nota **Re**. Più ci si discosta da 440, più l'affidabilità delle note diminuisce.
- **valore di default:** 440
- **durata:** la nota **La** viene suonata durante la durata dell'intervallo di tempo in secondi. Se un'altra nota come il **Si** viene suonata prima della fine dell'esecuzione di **La**, **La** viene fermata e **Si** la sostituisce
- se la durata viene omessa o se la durata = 0, la nota viene suonata a tempo indeterminato.
- se la durata è < 0, non viene suonata nessuna nota.
- **suggerimento:** per fermare una nota prima della sua fine (nota suonata a tempo indeterminato per esempio) senza sentirne una nuova, suonare una nota con durata = -1.
- **valore di default:** 1.

< accendere suono su scheda SD (numero) >



Riproduce un suono registrato sulla scheda SD.

Questi suoni sono stati creati con un sistema esterno e copiati sulla scheda SD.

I suoni devono essere chiamati **Px.wav**, dove **x** è il numero del suono da riprodurre.

Devono essere in 8 bit non firmati, 8 kHz e in formato mono.

numero: numero del suono registrato o copiato. Da 0 a 32767.

< registra suono (numero) >



Registra il suono raccolto dal microfono su una scheda SD inserita nel retro e genera un file audio chiamato **Rnumero.wav**. Questo suono è in formato 8 bit non firmato, 8 kHz e mono.

- **numero:** è possibile assegnare un numero ad una registrazione per avere più registrazioni.

- valore di default: 0
- limiti: da 0 a 32767.

< interrompere la registrazione del suono >



Interrompe la registrazione attraverso il microfono e salva il suono sulla scheda SD.

< riproduci suono (numero) >



- Riproduce un suono precedentemente registrato
- numero: corrisponde al numero del suono registrato (non al suono copiato sulla scheda SD)
- valore di default: 0.

5 Blocchi eventi

Nota: affinché gli script appesi sotto i blocchi degli eventi vengano eseguiti non è necessario che siano lanciati con la bandierina verde di Scratch per poter funzionare. La semplice presenza nell'area script è sufficiente.

< quando si preme il pulsante [opzione] >



Attiva l'esecuzione dei blocchi appesi sotto quando viene premuto uno dei 5 tasti Thymio.

- opzione: pulsante centrale | davanti | dietro | a sinistra | a destra
- valore di default: centrale.

Nota: per l'esecuzione dello script, non è necessario che venga lanciato con la bandierina verde Scratch.

< quando oggetto è rilevato [opzione] >



- Attiva l'esecuzione dei blocchi appesi sotto quando un oggetto viene rilevato da uno dei 9 sensori Thymio
- opzione: davanti | dietro | sotto
- valore di default: di fronte.

< quando nessun oggetto è rilevato [opzione] >



- Attiva l'esecuzione dei blocchi appesi sotto quando i sensori non rilevano nulla.
- opzione: davanti | dietro | sotto
- valore di default: di fronte.

< quando oggetto è rilevato [opzione 1] [opzione 2] >

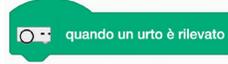


Attiva l'esecuzione dei blocchi appesi sotto quando un oggetto viene rilevato da uno dei 9

sensori Thymio.

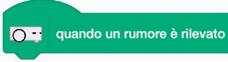
- opzione 1: anteriore | posteriore | inferiore
- valore di default: anteriore
- opzione 2: vicino | lontano. Con un sensore, l'opzione vicino corrisponde a circa 3 cm e l'opzione lontano a 6 cm.
- valore di default: vicino.

< quando un urto è rilevato >



Fa partire l'esecuzione dei blocchi che sono appesi sotto, quando Thymio è urtato. L'urto viene rilevato grazie all'accelerometro.

< quando un rumore è rilevato >



Fa scattare l'esecuzione dei blocchi appesi sotto quando Thymio capta un suono di intensità arbitraria. In pratica, il suono deve essere abbastanza forte per funzionare.

6. Blocchi logici

< pulsante [opzione] >



Restituisce **Vero** se viene premuto uno dei 5 pulsanti e **Falso** se non lo è.

- opzione: pulsante centrale | davanti | dietro | sinistra | destra
- valore di default: centrale.

7. Blocchi sensore

< sensori pross. orizz. [opzione] >



Restituisce un valore relativo all'intensità del segnale infrarosso ricevuto da uno dei sette sensori di prossimità orizzontale.

Varia tra 0 e 4000 circa. Più un oggetto è vicino, più alto è questo valore

- opzione: sinistra | centro a sinistra | centro a destra | destra | retro a sinistra | retro a destra

Nota: un'opzione può essere sostituita da un numero intero da 0 a 6 con la seguente corrispondenza: sinistra: 0 | centro a sinistra: 1 | centro: 2 | centro a destra: 3 | destra: 4 | indietro a sinistra: 5 | indietro a destra: 6

- valore di default: centrale.

< mostra tutti i sensori di pross. oriz. >

 mostra tutti i sensori di pross. oriz.

Restituisce la sequenza dei valori dei 7 sensori Thymio orizzontali.

Se non viene rilevato alcun ostacolo, viene visualizzato: 0 0 0 0 0 0 0

se il sensore centrale rileva qualcosa, può dare: 0 0 4318 0 0 0 0

Questo sensore è utile se si vuole sapere se viene rilevato o meno un ostacolo. Con il blocco

< **Sensori orizzontali di prossimità [opzione]** >, dovrebbero essere effettuati 5 test.

Nota: questo blocco ha una visualizzazione in Scratch come segue (casella di controllo)



< sensori terreno [opzione] >

 sensore terreno sinistra ▾

Restituisce un valore relativo all'intensità del segnale a infrarossi ricevuto da uno dei due sensori in basso.

Varia tra 0 e 1000 circa. Quando Thymio è posto su una superficie, più chiara è la superficie, maggiore è questo valore. Se Thymio è capovolto, visualizza 0.

• **opzione:** sinistra | destra

Nota: un'opzione può essere sostituita da una variabile intera da 0 a 6 con la seguente corrispondenza: sinistra: 0 | centro a sinistra: 1 | centro: 2 | centro a destra: 3 | destra: 4 | indietro a sinistra: 5 | indietro a destra: 6

• **valore di default:** sinistra.

< mostra tutti i sensori terreno >

 mostra tutti i sensori terreno

Restituisce la sequenza dei valori dei 2 sensori terreno di Thymio.

Se non viene rilevato alcun ostacolo, viene visualizzato: 0 0

Nota: questo blocco ha una visualizzazione in Scratch (casella di controllo).

< distanza [opzione] >

 distanza davanti ▾

Restituisce un valore che aumenta con la distanza da un oggetto che si trova nelle vicinanze. Non esiste una semplice relazione matematica tra questo valore e la distanza.

• **opzione:** anteriore | posteriore | inferiore

• **valori restituiti:**

• **Opzione frontale:**

- senza oggetti davanti, restituisce 190
- tutti i 5 sensori coperti, restituisce 0
- se c'è una misto di sensori nascosti e non ostacolati, restituisce un valore intermedio
- Infine, il valore indicato diminuisce con la distanza, ma è anche fortemente influenzato dal numero di sensori coinvolti e quindi dalle dimensioni dell'oggetto (o dal suo colore).

• **Opzione dietro:**

- senza oggetti dietro, restituisce 125
- con i 2 sensori coperti, restituisce 0 (o un valore molto basso)

- se solo uno viene nascosto, restituisce un valore vicino allo zero, l'altro sensore ha pochissima influenza.
- **Opzione sotto:**
 - sollevato, girato verso l'alto o di lato, restituisce 500
 - posato su un pavimento sottostante, restituisce 0. Se il pavimento è molto scuro, può restituire un valore elevato (> 400). Può essere utilizzato con successo su un pavimento con una variazione di grigio.

• **valore di default:** davanti.

< **angolo [opzione]** >

 angolo davanti ▾

Restituisce un valore che varia con l'angolo che un oggetto fa con l'asse avanti e indietro di Thymio.

- **opzione:** anteriore | posteriore | inferiore
- **valori restituiti:**

• **Opzione frontale:**

- senza oggetto davanti, restituisce 0
- se l'oggetto si trova sulla destra, restituisce valori negativi. Il valore aumenta con l'angolo (in valore assoluto)
- se l'oggetto si trova a sinistra, restituisce valori positivi. Il valore aumenta con l'angolo
- se l'oggetto è al centro, restituisce 0
- se l'oggetto è largo e oscura per esempio entrambi i sensori sul lato destro allo stesso tempo, il valore restituito è più grande che se fosse oscurato un solo sensore, anche se la posizione corrisponde ad un angolo reale più piccolo. Ad esempio, sensore centrale a destra: -25; sensore dritto: -34; entrambi i sensori: -59.

• **opzione dietro:**

- nessun oggetto dietro, restituisce 0
- i 2 sensori coperti, restituiscono 0 (o un valore molto basso)
- sensore sinistro: valori negativi, variabili a seconda della posizione e della distanza
- sensore destro: valori positivi, variabili a seconda della posizione e della distanza

• **Opzione sotto:**

- nessun oggetto sotto o se entrambi i sensori sono nascosti, restituisce 0.
- sensore sinistro: ritorna da 0 a 20 quando un oggetto si avvicina.
- sensore destro: ritorna da 0 a -20 quando un oggetto si avvicina
- se entrambi i sensori sono interessati, restituisce la somma algebrica dei 2 valori

• **Valore di default:** davanti.

< **inclinazione [opzione]** >

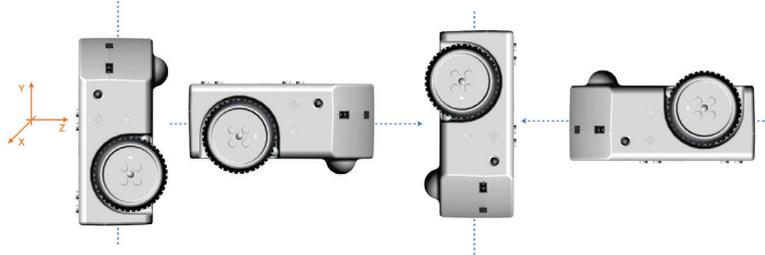
 inclinazione davanti-dietro ▾

Restituisce un valore che varia con le diverse inclinazioni del robot.

- **opzione:** davanti-dietro | sopra-sotto | sinistra-destra
- **valori restituiti:** per tutte le opzioni il valore varia tra -22 e 22 circa.

Una rotazione attorno ad un asse verticale (destra o sinistra) non modifica il valore restituito dal sensore perché restituisce una componente dell'accelerazione di gravità, verticale per definizione e la cui componente orizzontale è pari a zero.

Rotazione attorno all'asse sinistra-destra



Davanti-dietro	+22	0	-22	0
Sopra-sotto	0	+22	0	-22
Sinistra-destra	0	0	0	0

Rotazione attorno all'asse avanti-dietro



Avanti-dietro	0	0	0	0
Sopra-sotto	+22	0	-22	0
Sinistra-destra	0	+22	0	-22

< **livello sonoro** >

 livello sonoro

Restituisce il livello sonoro rilevato dal microfono.

- **limite:** da 0 a 255.

Nota: questo blocco ha una visualizzazione in Scratch (casella di controllo).

< odometria [opzione] >

odometria direzione ▾

Restituisce i parametri di movimento del robot rispetto ai parametri impostati dal blocco < **inizializza contachilometri** >.

- **Opzione: direzione | x | y**
 - valori restituiti
 - direzione: angolo di rotazione del robot da quando è stato inizializzato (a 90 per default). l'angolo viene contato positivamente se la rotazione è in senso orario
 - x: distanza percorsa da quando è stata inizializzata (a 0 per default). L'asse x è l'asse anteriore - posteriore. L'unità è la stessa del blocco < **avanzare** >.
 - y: distanza percorsa da quando è stata inizializzata (a 0 per default). L'asse y è l'asse sinistra - destra. L'unità è identica a quella del blocco < **avanza** >.

< velocità motori [opzione] >

velocità motori sinistro ▾

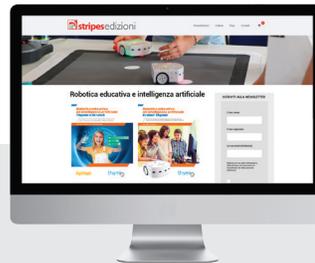
Restituisce il numero di giri effettivo di una delle 2 ruote.

- **Opzione: sinistra | destra**
 - valori restituiti : questa velocità può, per un momento, differire abbastanza fortemente dalla velocità richiesta dal blocco < **motori [opzione] alla velocità (velocità)** >, ma in media rimane vicina al valore desiderato.

Nota: non funziona con tutti i blocchi < **avanza...** > e < **ruota...** >

Scopri gli altri libri della collana

Resta informato sulle nuove pubblicazioni e scarica i materiali per le tue attività in classe.



www.stripesedizioni.it/roboticaeducativa



Licenza Open Education

Questa pubblicazione è distribuita sotto licenza Creative Common BY-SA-NC.

Crediamo nell'apertura delle risorse educative come strumento per un accesso libero e per quanto possibile gratuito all'istruzione, come diritto costituzionale. L'accesso alle risorse educative è una questione di giustizia sociale.

Le attività sono state progettate e distribuite in licenza Creative Common BY-SA, per promuovere l'adozione della robotica educativa nelle scuole con Thymio e Scratch, da parte dei seguenti autori: Joel Rivet e Didier Roy.

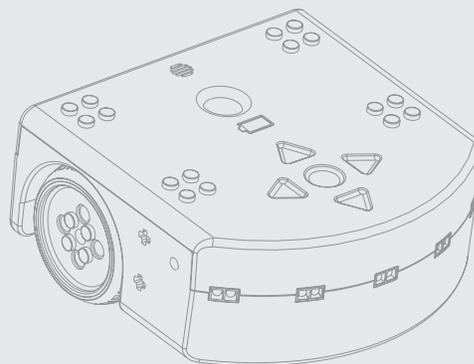
La pubblicazione è sostenuta anche grazie al contributo della Scuola Politecnica di Losanna (EPFL) in Svizzera.

Progetto grafico di Lorenzo De Manes e Paolo Rossetti.

Il logo Thymio è gestito dalla associazione elvetica senza scopo di lucro Mobsya. (www.mobysa.org)

Il software Thymio Suite è realizzato dalla Scuola Politecnica di Losanna (EPFL - Svizzera), dall'Istituto di Scienze e Tecnologie della Cognizione del Consiglio Nazionale delle Ricerche (ISTC-CNR - Italia) e dalla Associazione Mobsya (Svizzera). È reso disponibile in maniera completamente gratuita.

Scratch è un progetto del Lifelong Kindergarten Group dei Media Lab del MIT (USA). È reso disponibile in maniera completamente gratuita.



Editore **Stripes Coop. Sociale**

Via San Domenico Savio, 6 - 20017 - Rho (MI)
Tel. +39 02 931 6667 - Fax +39 02 935 070 57
P.IVA e C.F. 0963 5360 150
www.stripesedizioni.it

Stampato presso
Fabbrica dei segni Coop. Sociale
Via Baranzate, 72/74
Novate Milanese (MI)
Dicembre 2020
Edizione - BASE-IT-IT-12-20-AA

Educazione aperta e collaborazione internazionale

SCRA-IT-IT-12-20-AA

Questo secondo volume della collana Robotica educativa e Intelligenza Artificiale è dedicato agli insegnanti che insegnano il linguaggio di programmazione Scratch (giunto alla sua versione 3) ed alle estensioni per programmare il robot Thymio disponibili con il software Thymio Suite. Il robot Thymio nasce al Politecnico di Losanna con uno spirito unico nel panorama dei robot educativi: una filosofia completamente "Open" con una attenzione ed una sensibilità per il mondo delle Scuole. Scratch è un linguaggio di programmazione del Lifelong Kindergarten Group dei Media Lab del MIT (Massachusetts Institute of Technology - Boston USA) reso disponibile in maniera completamente gratuita ed estremamente popolare in ambito scolastico.

Con Scratch e Thymio puoi programmare storie interattive, giochi e animazioni interagendo con un robot pensato per le attività didattiche. Scratch insegna ai giovani a pensare in maniera creativa, a ragionare in modo sistematico e a lavorare in maniera collaborativa — queste sono tutte capacità essenziali per chi vive nel 21mo secolo. Thymio interagisce grazie ai suoi sensori e attuatori e fornisce feedback visivi istantanei su quanto i sensori ricevono dal mondo esterno per rendere "trasparente" il suo funzionamento.

Questa pubblicazione è il frutto di una scelta etica di autore, editore, stampatore. Paolo Rossetti anima un'associazione di promozione sociale nel territorio dove vive - Associazione La Forma del Cuore - i Fondatori e i Soci della Cooperativa Stripes sperimentano quotidianamente i benefici della robotica educativa presso lo Stripes Digitus Lab (all'interno di MIND Milano Innovation District) e la Cooperativa Fabbrica dei Segni realizza questo libro anche con il contributo di persone svantaggiate.

Questo libro è anche reso disponibile online, sempre in versione PDF fotocopiabile per gli insegnanti, sulla piattaforma degli insegnanti di robotica educativa roteco.ch, sempre a titolo gratuito. Entra a far parte della comunità Roteco dove troverai colleghi con cui scambiare attività didattiche ed educative sulla robotica, l'intelligenza artificiale ed il pensiero computazionale, il coding e le materie MINT (Matematica, Informatica, scienze Naturali, arte e Tecnologie).

EPFL



LEARN
Center for Learning Sciences

Euro 22,00

www.stripesedizioni.it/roboticaeducativa

ISBN 978-88-88952-54-3



9 788888 952543

