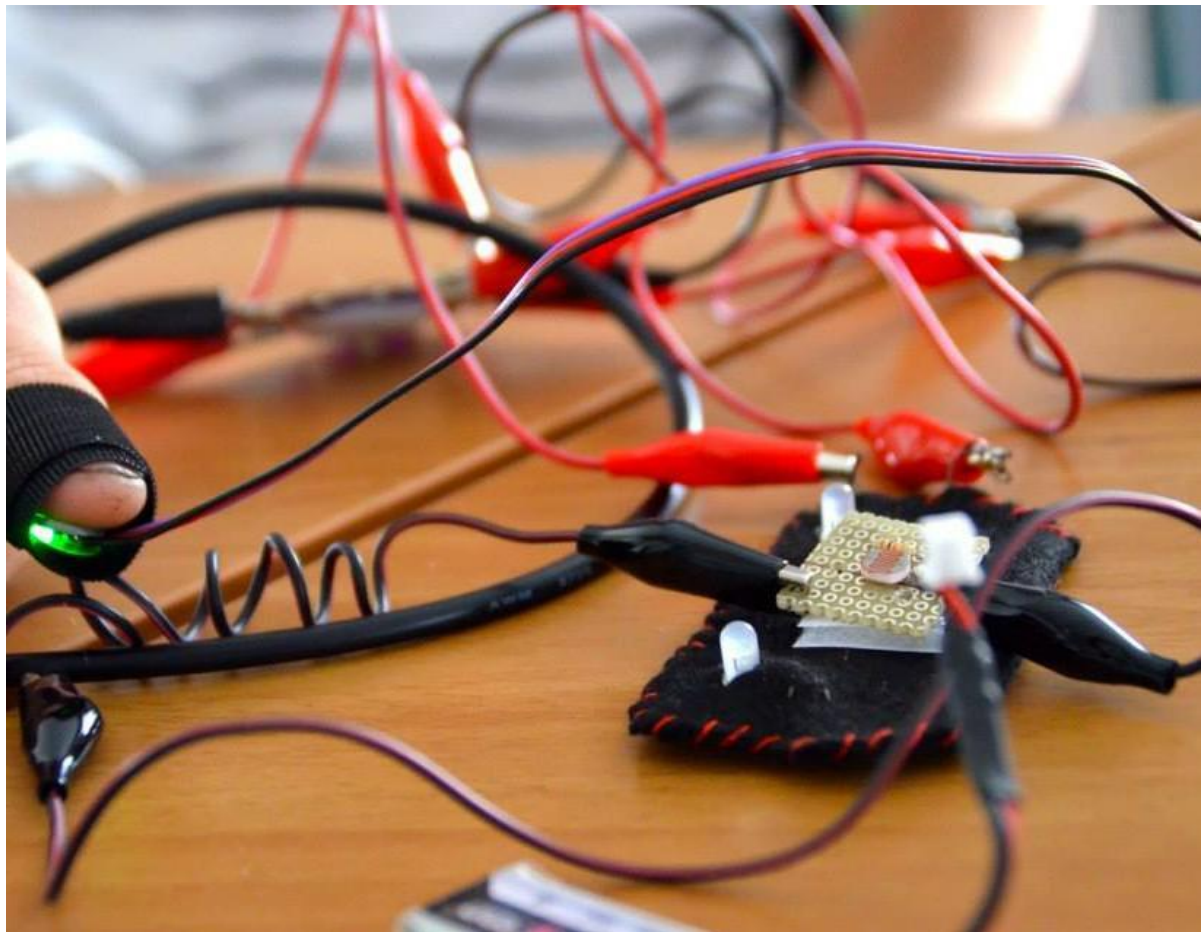


IOT per scopi educativi e scolastici

L'obiettivo di questo manuale è introdurre le diverse metodologie, gli Scenari di Apprendimento e le attività educative sulla programmazione e l'IoT nell'istruzione.



To thrive, they must learn to design innovative solutions to the unexpected problems that will undoubtedly arise in their lives. Their success and satisfaction will be based on their ability to think and act creatively. Knowledge alone is not enough: they must learn how to use their knowledge creatively

- Mitchel Resnick, MIT Media Lab

Introduzione all'Internet of things e ai dispositivi indossabili

In questo manuale possiamo imparare come usare alcuni semplici editor online per programmare e interagire con il mondo esterno con piccoli sensori. L'Internet of things è in crescente aumento sul mercato attuale, si pensi ai termosifoni e agli smart watch.

Nel campo della formazione sarà importante parlare di questa tecnologia in quanto è un'attività che coinvolge gli studenti e perchè vede l'applicazione della teoria delle 4P.

Mitchel Resnick dell'MIT Lifelong Learning Lab (MediaLab) ha introdotto questa metodologia delle 4 P (Project, Peer, Play and Passion) e questa si adatta molto bene alle attività formative dell'IoT. Inoltre in questo manuale presentiamo la possibilità di creare alcuni programmi con diverse piattaforme commerciali, tutte collegate dall'editor online scratch3, con cui è possibile controllare alcuni sensori.

Negli Scenari di Apprendimento utilizzeremo anche programmi diversi come Snap per Arduino o Makecode. Tutti questi utilizzano software gratuite compatibili con le più importanti piattaforme robotiche come Lego, Microbit, Arduino and Raspberry PI.

Introduzione ad Arduino e a schede simili

Scegliamo di presentare solo Arduino e le piattaforme compatibili perchè sono open source e nel progetto europeo noi consideriamo etico l'utilizzo di piattaforme economiche e open source come Arduino and Elegoo.

Questa scheda è totalmente compatibile con una moltitudine di sensori e attuatori.

Arduino è la piattaforma open source più conosciuta che cambia il mondo dell'istruzione, grazie al basso costo e la semplicità di prototipazione e programmazione.

Un tipico microcontrollore è composto da 8 bit con diversi chip dalla famiglia Mega AVR. In ogni schedina vi sono input Digitali e Analogici e output Digitali e Analogici.

Software online gratuito

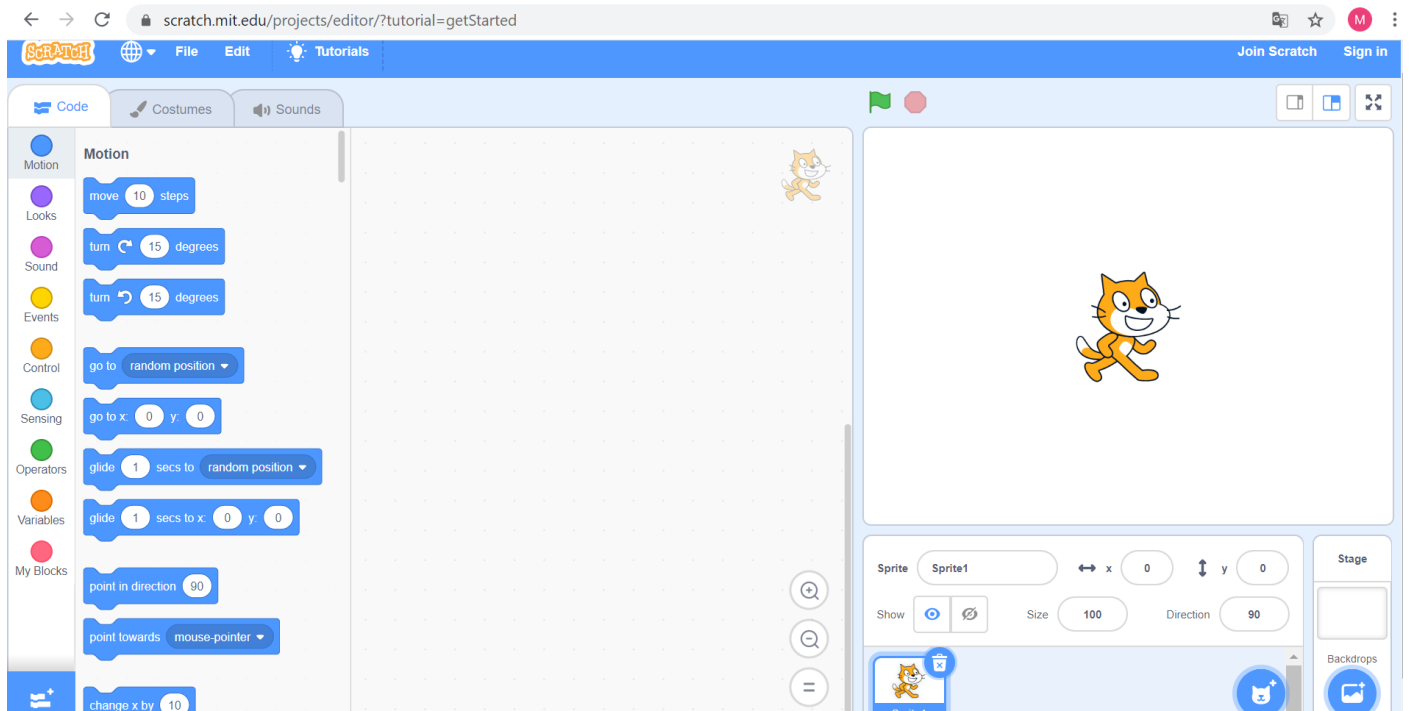
In questo progetto scegliamo di utilizzare solo software gratuiti per programmare i nostri oggetti. Il più conosciuto è Scratch, attualmente aggiornato alla versione Scratch3.

Scratch

Scratch è un editor a programmazione grafica/visiva sviluppato da MIT Media Lab. Scratch è nato nel 2006 e attualmente è utilizzato nella maggior parte delle scuole del mondo. È disponibile in più di 70 lingue.

Scratch 2 è disponibile anche in modalità offline. Puoi scaricarlo al link <https://scratch.mit.edu/download>. Scratch 2 non è disponibile nella versione tablet.

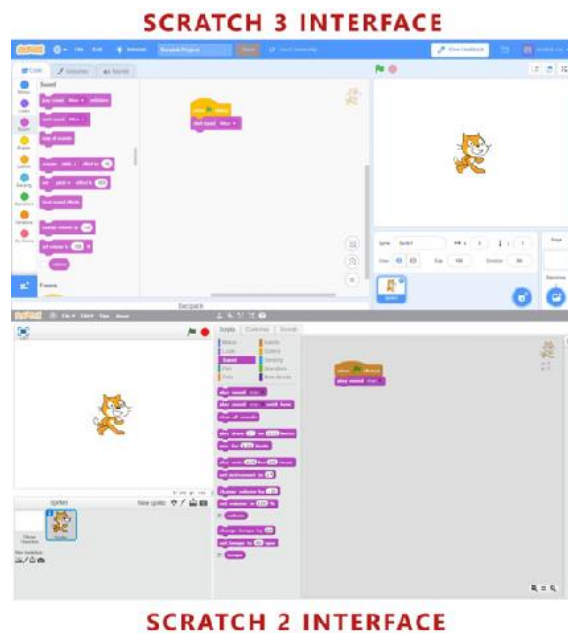
Queste sono le principali differenze tra Scratch 2 e 3. Entrambe le versioni di Scratch possono lavorare online attraverso il browser internet sul pc. Scratch 3 è stato creato in HTML5 in modo che potesse lavorare anche su tablet Android o iPad, ma non presenta la versione al contrario di Scratch 2 che è stato creato in Flash. Entrambi i programmi possono interagire con dispositivi esterni ma solamente su computer in quanto hanno bisogno che venga installato un piccolo programma. Scratch 2 può controllare Lego WeDo 1 and 2 and Picoboard. Scratch 3 può controllare Lego WeDo 2 e Lego Mindstorm EV3, Microbit, ha inoltre delle funzionalità extra come trasformare un testo in discorso in diverse lingue e tradurlo. È possibile costruire nuove estensioni così che in futuro vi saranno nuove funzionalità. Vi saranno altre nuove funzionalità comuni alle due piattaforme come la penna per disegnare, strumenti musicali e video sensoriali.



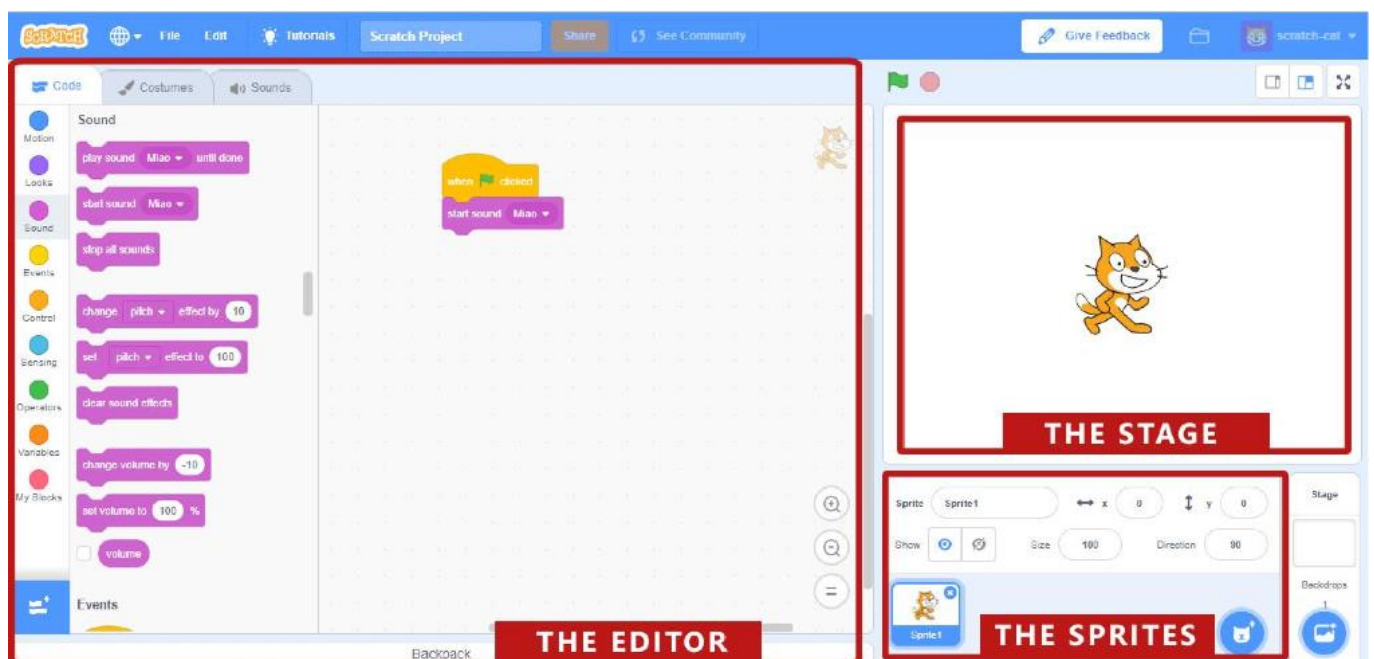
Differenze tra Scratch 2 e Scratch 3		
	Scratch 2	Scratch 3
Versione offline (Windows, OSX)		
Versione online	(solo per computer)	(per computer e tablet)
Device esterni		
Lego WeDo 1	X	
Lego WeDo 2	X	X
Picoboard	X	
Lego Mindstorm EV3	X	X
Microbit		X
Funzionalità esterne		
Da testo a discorso in diverse lingue	X	X
Penna per disegnare	X	X
Strumenti musicali		X
Video sensoriali		X
Traduttore testo		X

Entrambe le versioni permettono di salvare e condividere i progetti Scratch sulla Scratch Community. I file sono compatibili con entrambe le versioni. Da Gennaio 2019 la sola versione online sarà Scratch 3 e la sola versione offline sarà Scratch 2.

Interfaccia



La nuova interfaccia di Scratch 3 è mostrata nell'immagine sottostante.



È diviso in tre principali aree:

Lo stage è l'area più importante dove il programma scritto 'prenderà vita'.

L'area dello sprite, in questa area ci sono tutti i componenti (sprite), che fanno parte dello stage.

L'area editor, questa comprende tre tipi differenti di editor:

1. **L'editor di codifica**: contiene l'elenco dei blocchi disponibili e tutti i blocchi utilizzati per descrivere il comportamento del rispettivo sprite. È importante capire che ogni sprite ha i suoi blocchi, quindi quando scegliamo diversi sprite dall'area in cui è possibile sceglierli e cambiarli, i blocchi nell'area dello script possono variare.
2. **L'editor dei costumi**: permette di disegnare e modificare l'aspetto degli sprite. Ogni sprite ha i suoi costumi.
3. **L'editor dei suoni**: permette di registrare e modificare i suoni da utilizzare nel programma.

Snap per Arduino

Snap per Arduino è una modifica del blocco programma Snap!, creato dall'Università della California a Berkeley. Grazie a Snap (che è in continuo sviluppo) possiamo programmare semplicemente tutte le schede Arduino.



Dal sito ufficiale, le caratteristiche di Snap4Arduino sono:

- Programmazione a blocchi, dinamica, in tempo reale, concomitante, parallela
- Supporto per tutte le schede Arduino
- Utilizza il firmware standard Firmata

- Pinout autoconfigurabili e astrazioni hardware di alto livello
- È possibile interagire con più schede contemporaneamente
- Versioni desktop per i tre principali sistemi operativi
- Versione online che può connettersi alle schede Arduino attraverso il plugin Chrome/Chromium
- Licenza software gratuita attraverso Affero GPLv3
- Trasformazione di semplici scripts in programmi Arduino
- Protocollo HTTP per il controllo remoto e lo streaming in tempo reale di Snap!
- Versione per GNU/Linux integrati

Come installare ogni software

Per installare Scratch 3: <https://scratch.mit.edu>

Per installare Snap4Arduino: <http://snap4arduino.rocks>

Focus su Scratch 3

Scratch è progettato in particolare per bambini con età compresa tra gli 8 e i 16 anni, ma il tool è facilmente accessibile a programmatori principianti.

Scratch è utilizzato da più di 150 paesi ed è disponibile in più di 40 lingue.

Scratch è utilizzato come un'introduzione al linguaggio in quanto è relativamente semplice creare programmi interessanti e le capacità acquisite possono essere applicate ad altri linguaggi di programmazione come per esempio Python e Java.

Category	Notes	Category	Notes
Motion	Moves sprites, changes angles and changes X and Y values.	Sensing	Sprites can interact with the surroundings the user has created
Looks	Controls the visuals of the sprite; attach speech or thought bubble, change of background, enlarge or shrink, transparency, shade	Operators	Mathematical operators, random number generator, and-or statement that compares sprite positions
Sound	Plays audio files and effects. Programmable sequences are now available as an extension category named "Music".	Variables	Variable and List usage and assignment
Events	Contains event handlers placed on the top of each group of blocks	My Blocks	Custom procedures (blocks).
Control	Conditional if-else statement, "forever", "repeat", and "stop", etc.		

In aggiunta, Scratch contiene le seguenti estensioni:

- Musica
- Penna

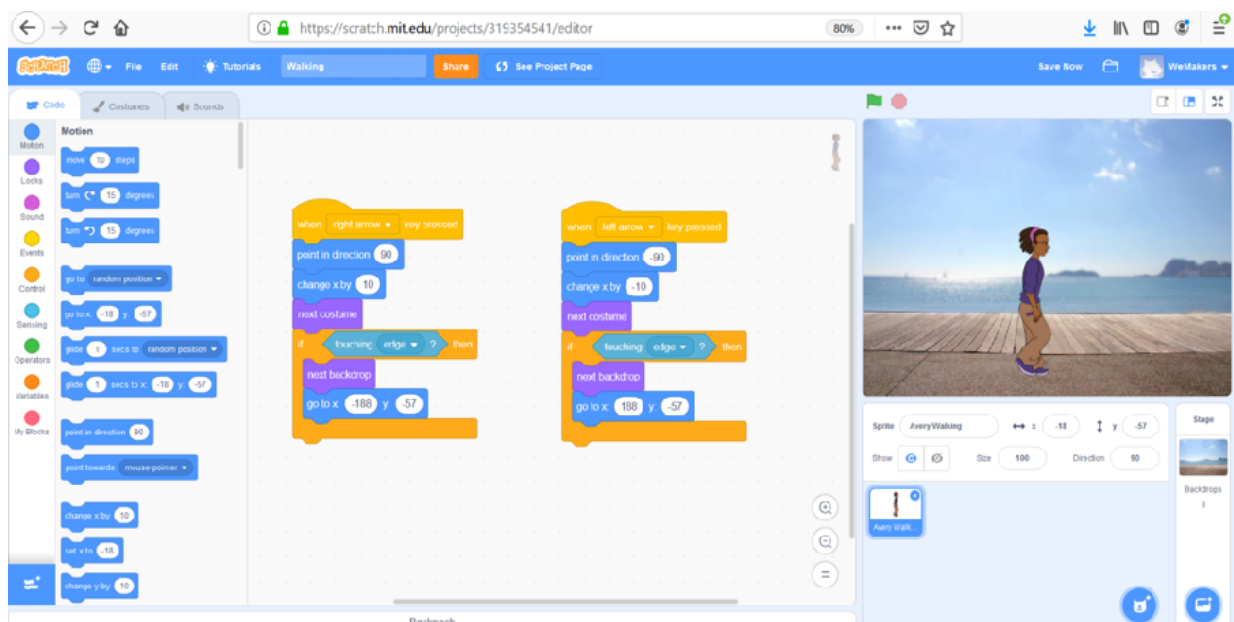
- Video sensoriali
- Da testo a discorso
- Traduzione
- Makey Makey
- micro:bit
- LEGO MIDSTORMS EV3
- LEGO BOOST
- LEGO Education WeDo 2.0
- Go Direct Force & Acceleration

Applicazione 1: uno sprite che cammina (freccia sinistra/destra)

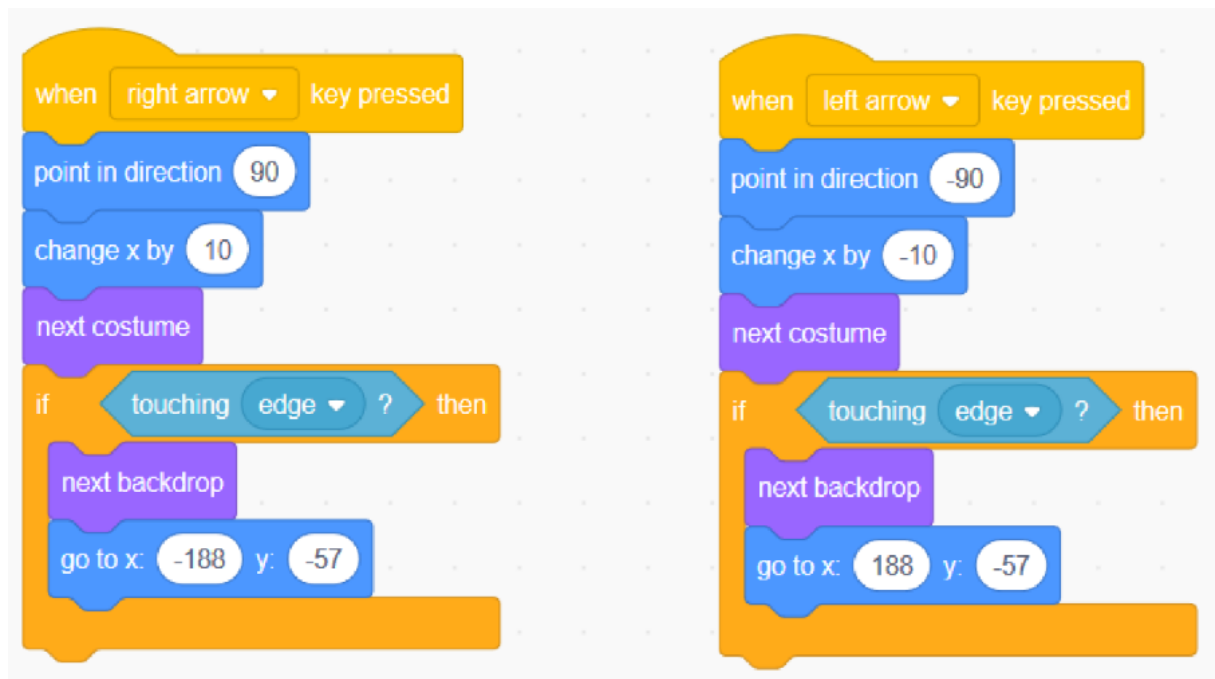
La seguente applicazione ha un personaggio (sprite) controllato da sinistra a destra, il quale cammina in tre o più sfondi.

Come effettuare la programmazione step per step:

1. Scegli uno sprite (preferibilmente uno che possiede le caratteristiche del cammino/volo/nuoto)
2. Scegli tre o più scenari/sfondi



3. Scrivi gli script rappresentati nella immagine seguente
4. Utilizza le frecce destra e sinistra per testare il risultato



Obiettivi proposti

Obiettivo 1: Fai in modo che lo sprite salti e avanzi quando il pulsante che indica la freccia in alto viene premuto.

Obiettivo 2: Create un gioco:

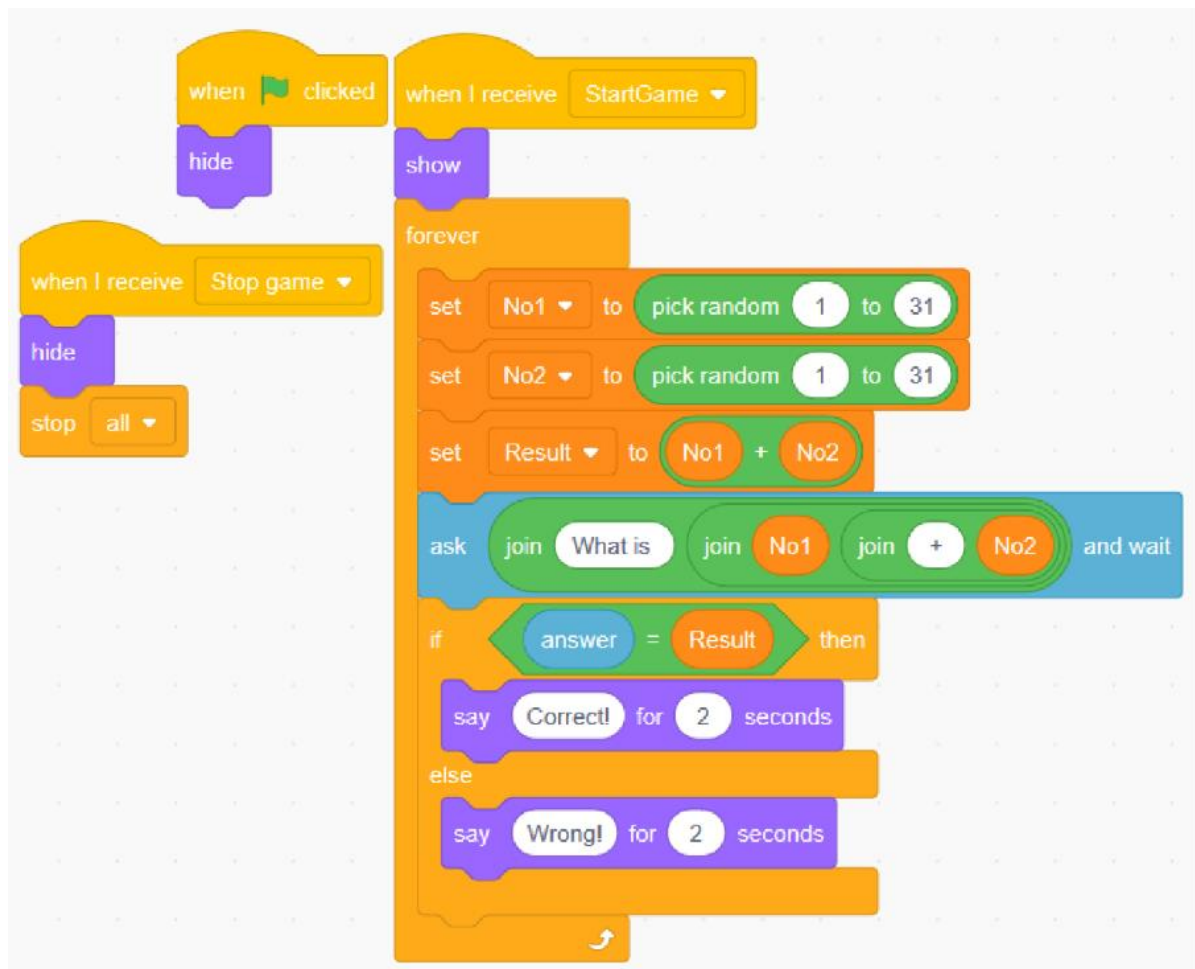
1. Controlla 1 sprite e una rana che avanza in modo randomizzato (varia il solo parametro x)
2. Lo sprite non deve toccare la rana (deve saltare oltre la rana)
3. Ha 3 vite
4. Se tocca la rana perde una vita
5. Il gioco finisce quando perdi tutte le vite

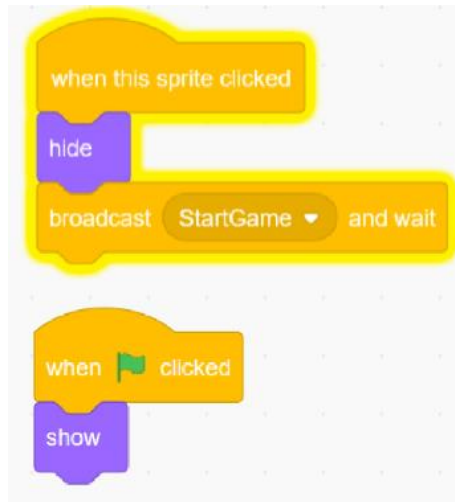
Puoi aggiungere una parte di codice appropriato per fare in modo che lo sprite si trasformi per 2 secondi in rana o in un altro animale quando tocca la rana.

Obiettivo 3: Il gioco genera in modo randomizzato finestre aggiuntive e proietta *Corretto* o *Sbagliato* ad ogni risposta. Ha tre sfondi e tre sprites (2 bottoni e uno sprite che fa una domanda e fornisce la risposta).



Gli script per ogni sprite e per ogni stage sono:





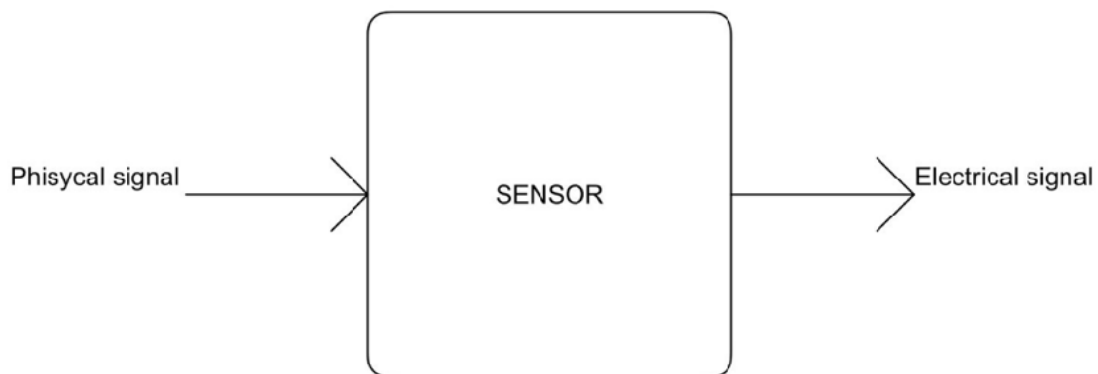
Obiettivo 4: Aggiungi due variabili per il numero totale di domande e il numero totale di risposte corrette. Alla fine fai in modo che questi due valori siano visibili.



Obiettivo 5: Crea il tuo gioco a quiz con 2-3 domande.

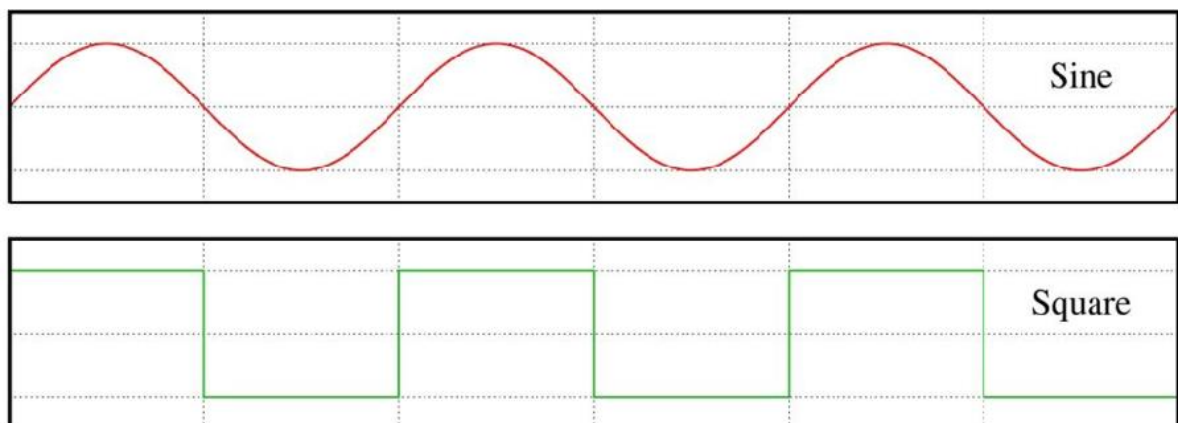
Sensori: una breve panoramica

Un sensore è un dispositivo che è utilizzato nell'elettronica per determinare cambiamenti nell'ambiente di un parametro fisico e inviare questo, codificato in un segnale elettrico, agli altri dispositivi in modo tale che possa essere elaborata e analizzata.



Il segnale elettrico può essere analogico o digitale. Analogico significa che il segnale varia con continuità, fornendo tutti i valori compresi tra il valore massimo e quello minimo. Invece digitale significa che il segnale può assumere solo un numero limitato di valori, tipicamente due: livello basso o livello alto.

Nella figura sottostante puoi vedere due esempi di segnale: un'onda sinusoidale e un'onda quadra. La prima assume tutti i valori dal basso verso l'alto, perciò è un valore analogico. La seconda assume solo due valori, passando dallo stato basso a quello alto e viceversa, perciò è un segnale digitale.



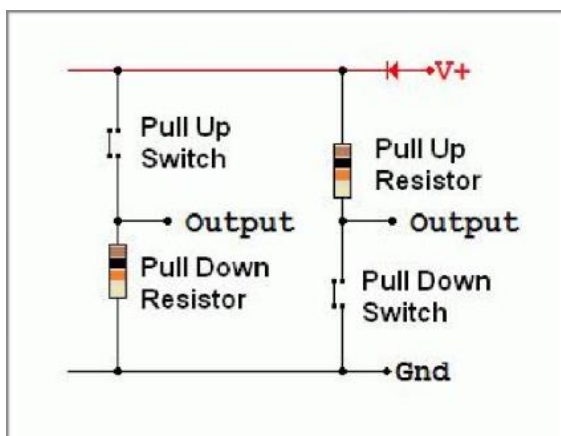
Un esempio di segnale analogico è la temperatura: può assumere tutti i valori dallo zero assoluto fino all'infinito. Il tipico esempio di segnale digitale è lo stato di un pulsante: questo può essere solamente acceso o spento. Il tipico sensore analogico è la variabile resistore (per esempio il potenziometro utilizzato per alzare ed abbassare il volume dei nostri amplificatori stereo), e quello digitale è il bottone, che vedremo nella successiva sezione.

Sensore di tatto

Il più semplice sensore che può essere costruito è il sensore tattile. Esistono diversi tipi di sensore tattili, ma il più semplice che possiamo utilizzare è il bottone e l'interruttore.

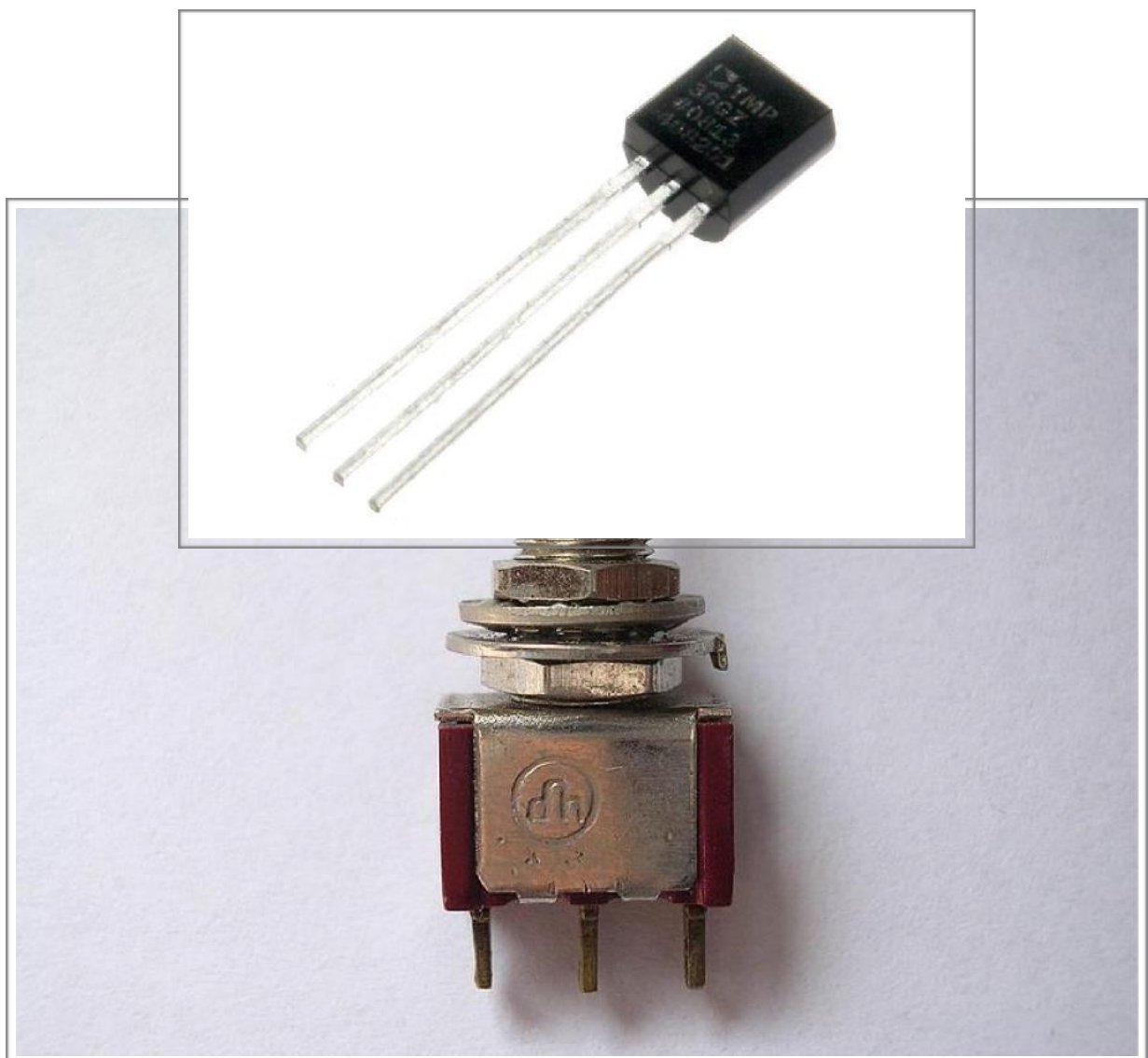
Un bottone è solamente un dispositivo elettromeccanico che è in grado di chiudere o aprire un circuito elettrico. Quando il circuito è chiuso la corrente elettrica può fluire, altrimenti quando è aperto la corrente non può fluire. Mettendo un bottone in un circuito apposito con un resistore, possiamo ottenere un circuito che può fornire solamente due valori di voltaggio, alto o basso, corrispondenti allo stato del bottone. Potete vedere lo schema sottostante per le connessioni. Potete mettere la resistenza e l'interruttore in due modi. La prima, con la “resistenza di pull down” offre un valore basso in uscita quando l'interruttore è aperto e un valore alto quando è chiuso.

Il “resistore di pull up” fornisce il comportamento contrario. Per esempio noi utilizziamo bottoni e interruttori a casa nostra per accendere e spegnere la luce, o in un ascensore per scegliere il piano giusto.



A questo link potete trovare maggiori informazioni a proposito del bottone e di una scheda programmabile open source: <https://www.arduino.cc/en/Tutorial/Button>

Esistono altri sensori tattili, e i più conosciuti sono i sensori capacitivi. La loro tecnologia è utilizzata nei nostri dispositivi touch screen. Misurano la variazione della capacità di un condensatore a causa della presenza umana. Alla fine, abbiamo un device che offre due livelli di uscita, come il pulsante.

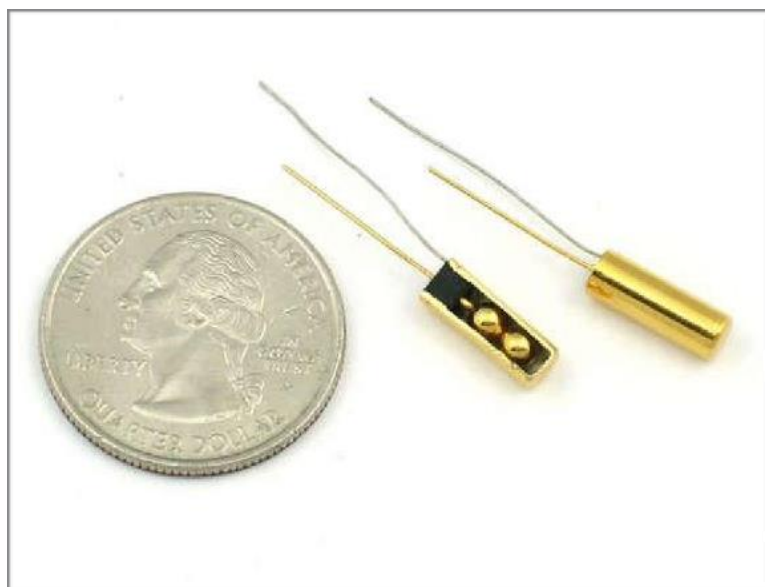


Sensore di tilt

Il sensore di tilt è utilizzato per determinare se un oggetto è inclinato in una o più direzioni. Il più semplice sensore di tilt è l'interruttore a mercurio: è composto da una lampadina vuota, contenente due contatti e una piccola quantità di mercurio, libera di muoversi. Quando la lampadina è inclinata il mercurio si muove e quando raggiunge i contatti chiude il circuito. Il mercurio è spesso sostituito con una sferetta di metallo. Utilizzando più contatti è possibile creare sensori di tilt multi-asse.

Un sensore di tilt è utilizzato, per esempio, in alcuni yo-yo che emettono luce, i quali lampeggiano quando giochi o in semplice sensori di pitch-roll in alcuni veicoli.

A questo link potete trovare un esempio con un sensore di tilt a singolo asse e una scheda programmabile: <https://learn.adafruit.com/tilt-sensor/using-a-tilt-sensor>

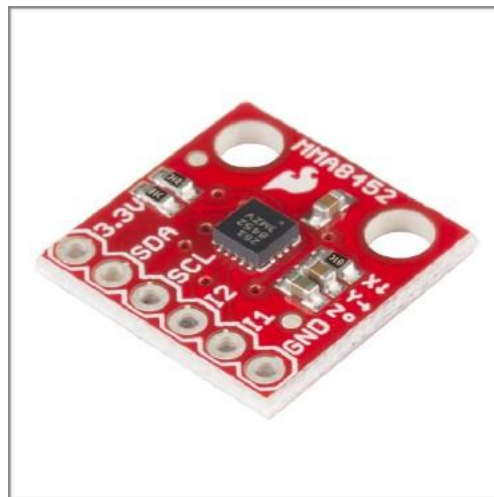


Accelerometro

L'accelerometro è un dispositivo utilizzato per misurare l'accelerazione. Lavora per mezzo dell'inerzia fisica di una massa. Concettualmente un accelerometro è composto da una massa connessa ad una molla. Quando l'accelerometro è soggetto a una accelerazione la massa si muove rispetto al contenitore, proporzionalmente al modulo dell'accelerazione. Misurando lo

spostamento, si ottengono informazioni sull'accelerazione a cui sono sottoposti i dispositivi. L'accelerazione viene poi codificata in qualche modo in un segnale elettrico. Gli accelerometri sono utilizzati ad esempio nei sistemi di navigazione per ottenere la posizione relativa riferita ad un punto zero, nel nostro smartphone come dispositivi di input, per ruotare lo schermo, per giocare a qualche gioco, come contapassi e così via, o in alcuni computer con hard disk magnetico come rilevatori di caduta per evitare problemi al disco rigido, impostando la testina di lettura in una posizione sicura.

A questo link potete trovare un esempio di utilizzo di un accelerometro analogico a 3 assi, che fornisce tre tensioni proporzionali alle accelerazioni, con una scheda programmabile: <https://www.arduino.cc/en/Tutorial/ADXL3xx>

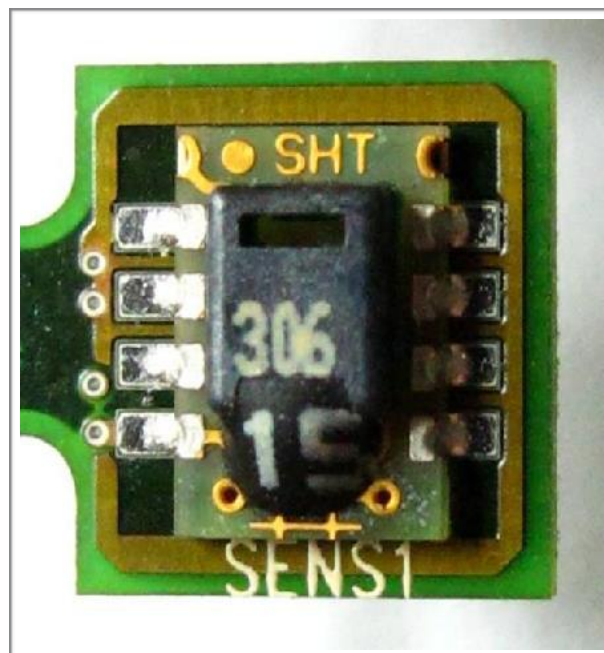


Sensore di umidità

Il sensore di umidità misura l'umidità in un mezzo, come ad esempio l'aria o il terreno. Generalmente lavorano misurando la variazione di capacità e o la resistenza dovuta alla variazione della quantità di acqua nel mezzo. I sensori di umidità sono usati per esempio in meteorologia, anche nella nostra stazione meteorologica portatile, e nel sistema di irrigazione automatica che permette di aprire l'acqua solo quando il terreno è asciutto.

A questo link potete trovare un esempio con una schedina open source e un comune sensore di umidità (che può anche misurare la temperatura):

https://www.tutorialspoint.com/arduino/arduino_humidity_sensor.htm



Sensore di temperatura

Un sensore di temperatura converte la temperatura in un segnale elettrico. Ciò è possibile grazie a diversi fenomeni fisici, che dipendono dal tipo di sensore. Ad esempio il termistore fornisce una variazione della sua resistenza in funzione di una variazione della temperatura, e la termocoppia fornisce, per l'effetto termoelettrico,

una tensione proporzionale alle differenze di temperatura.

I sensori di temperatura sono utilizzati in meteorologia, nei termometri medici, nelle CPU di computer ecc.

Qui si può vedere un esempio di lettura della temperatura utilizzando una scheda programmabile: <https://learn.adafruit.com/tmp36-temperature-sensor/using-a-temp-sensor>

Output: una breve panoramica

Un output è un elemento che può agire sul mondo reale utilizzando diversi modi di interazione. La più importante uscita è quella che utilizza il movimento, le luci o il suono per interagire con gli utenti e con l'ambiente.

LED

Se volete inserire LED nelle vostre classi potete utilizzare diversi metodi che dipendono dalla tipologia della vostra scuola. In questo manuale mostreremo generalmente tutte le componenti elettroniche di cui avete bisogno così che anche insegnanti di diverse discipline potranno utilizzarli senza problemi.

Prima di definire un componente è necessario collegare la creazione di un componente alla vita del suo inventore. L'inventore del LED è Nicholas Holonyak Jr, un americano di origini russe, che ha sviluppato il primo LED già nel 1962, era in grado di emettere solo la luce rossa, oggi è possibile acquistare LED di colori molto diversi. Grazie ad Asaki, Amano e Nakamura, ad esempio, sono stati creati i LED blu, questa scoperta è così importante che gli ha fatto vincere il premio Nobel.



Un LED è un'uscita apparentemente molto semplice che permette di emettere la luce di diversi colori in modo affidabile, dura a lungo ed è a basso costo. Oggi i LED sono utilizzati per molte applicazioni, non solo in ambito commerciale ma anche in ambito domestico.

Il principio di funzionamento si basa su elettroni che occupano alcuni vuoti in un materiale semiconduttore e rilasciano energia, o i fotoni. Le diverse frequenze, e anche i diversi colori, emessi dipendono dai materiali. Il LED ha bisogno di corrente costante per essere alimentato, questo è il motivo per cui va sempre con una resistenza che permette di controllare il flusso elettrico che raggiunge il diodo.

Altri esercizi:

Ecco una lista di alcuni esercizi aggiuntivi (che non sono trattati in questo manuale) che potranno aiutare i vostri studenti per allenarsi e migliorare le loro conoscenze, sarà utile anche per rendere più semplice la valutazione dell'insegnante:

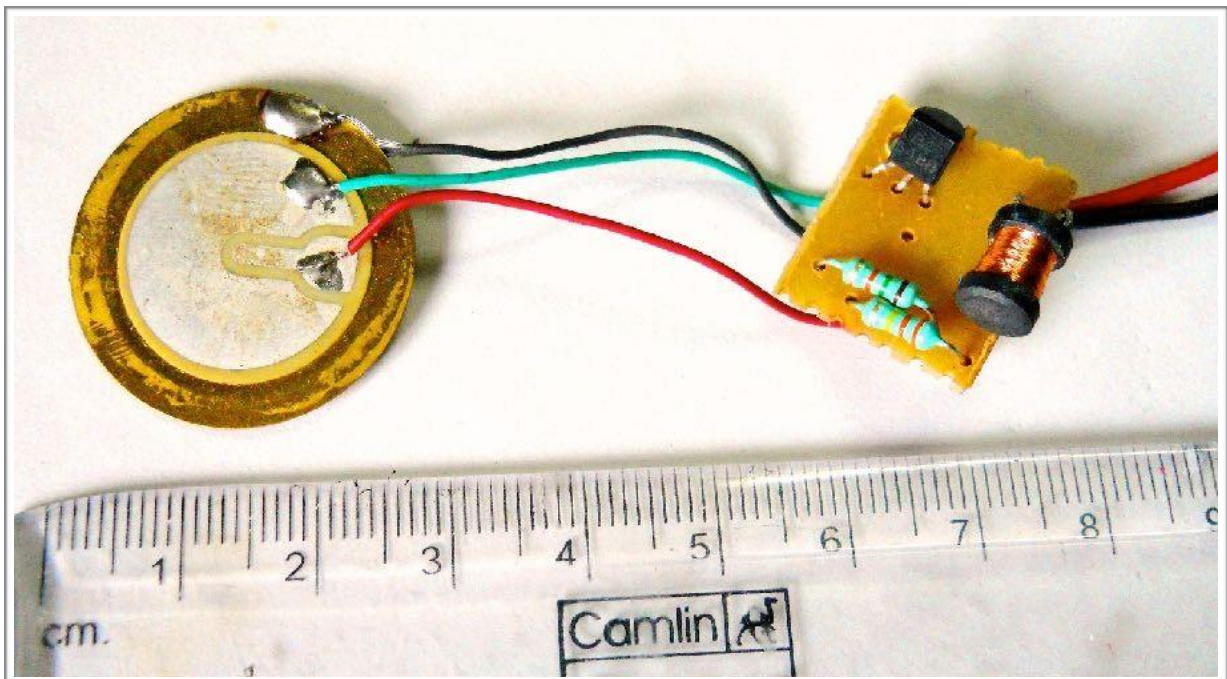
Cambiare il tempo di lampeggio del diodo, da 1 a 3 secondi.

Cambiare il pin digitale (not 0 and 1'1!) dove il LED è connesso e cambiare il programma in modo opportuno.

Cambiare il tempo di lampeggio in questo modo: LED acceso per 1.5 secondi- LED spento per 2/3 secondi.

Buzzer

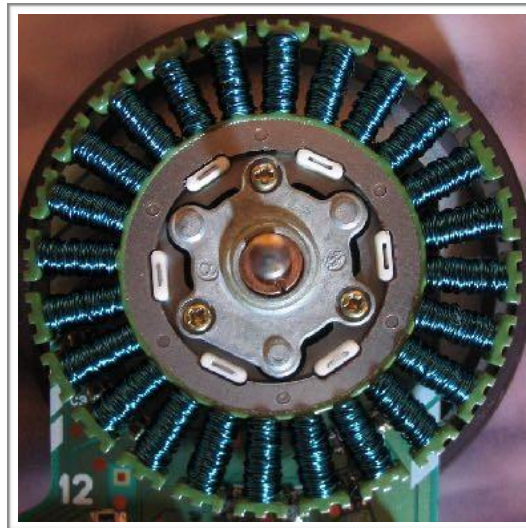
In primo luogo potete utilizzare Arduino per emettere suoni con diverse tonalità e lunghezze. Quindi potete parlare di musica e delle canzoni e musiche preferite dagli studenti. Inoltre è possibile guardare una parte di "Close Encounters of the Third Kind" di S. Spielberg, quella in cui gli umani comunicano con gli alieni attraverso 5 note musicali (l'interpretazione dello scienziato che fa iniziare questa scena è quella di Francois Truffaut). Ovviamente potete far fare agli studenti la loro music. In questo modo è possibile coinvolgere gli studenti ad usare la codifica in modo creativo. Creando un'introduzione su come emettere le note con il buzzer, potete programmare il software in modo che impari a suonare canzoni famose.



Motori

I motori permettono di muovere per la prima volta il robot, il primo movimento del robot sarà molto semplice e senza senso, il robot andrà avanti all'infinito senza curarsi degli ostacoli e del tempo di funzionamento. Nel mondo Arduino il ponte-H è un transistor che permette di controllare in modo simultaneo due motori DC. Senza il ponte-H non è possibile che il robot si muova. Il ponte-H infatti è una sorta di

batteria che gestisce l'energia del robot. I motori contenuti nei più famosi kit Arduino sono motori DC senza sensori. Ecco perchè non è possibile gestire la rotazione del motore ma solamente la corrente erogata.



“Every maker of video games knows something that the makers of curriculum don't seem to understand. You'll never see a video game being advertised as being easy. Kids who do not like school will tell you it's not because it's too hard. It's because it's--boring”

— Seymour Papert

Design Challenge

In questo capitolo presentiamo alcune sfide di progettazione guidate con diversi livelli di difficoltà e diversi programmi hardware e software da utilizzare.

Ogni Design Challenge è presentata come una sfida da risolvere con la vostra classe, utilizzando alcune metodologie didattiche:

- 4 P: Play, Passion, Peer and Project
- PBL: Project Based Learning
- Flipped Classroom

Qui alcune brevi definizioni per ogni metodologia che applicheremo nelle nostre sfide didattiche:

4P: Project, Play, Passion and Peer

Mitchel Resnick introduce con il libro *Kindergarden: LifeLong learning* il concetto di studente creativo.

Uno studente creativo ha bisogno di risolvere il problema focalizzandosi sul processo. A scuola ovviamente non è possibile risolvere il problema in modo drastico ma possiamo imparare uno o più metodi che potrebbero permettere di imparare.

Progetti:

Un progetto è un processo che può risolvere una serie di problemi che stanno compinando il problema che vogliamo risolvere.

Non abbiamo bisogno di un progetto concreto, possiamo anche utilizzare la teoria per trovare una proposta per migliorare uno o più problemi.

Peer:

La collaborazione è fondamentale nel processo di apprendimento, insegnamento per i nostril colleghi, condividere informazioni e ascoltare i nostril colleghi, compagni di

classe ci permettono di migliorare le nostre conoscenze e la nostra consapevolezza dei nostri amici e ascoltatori. Grazie a questa parte possiamo sviluppare empatia.

Passione:

Questo significa Impegno. Possiamo lavorare con sensazioni positive dei nostri studenti quando stiamo utilizzando la tecnologia.

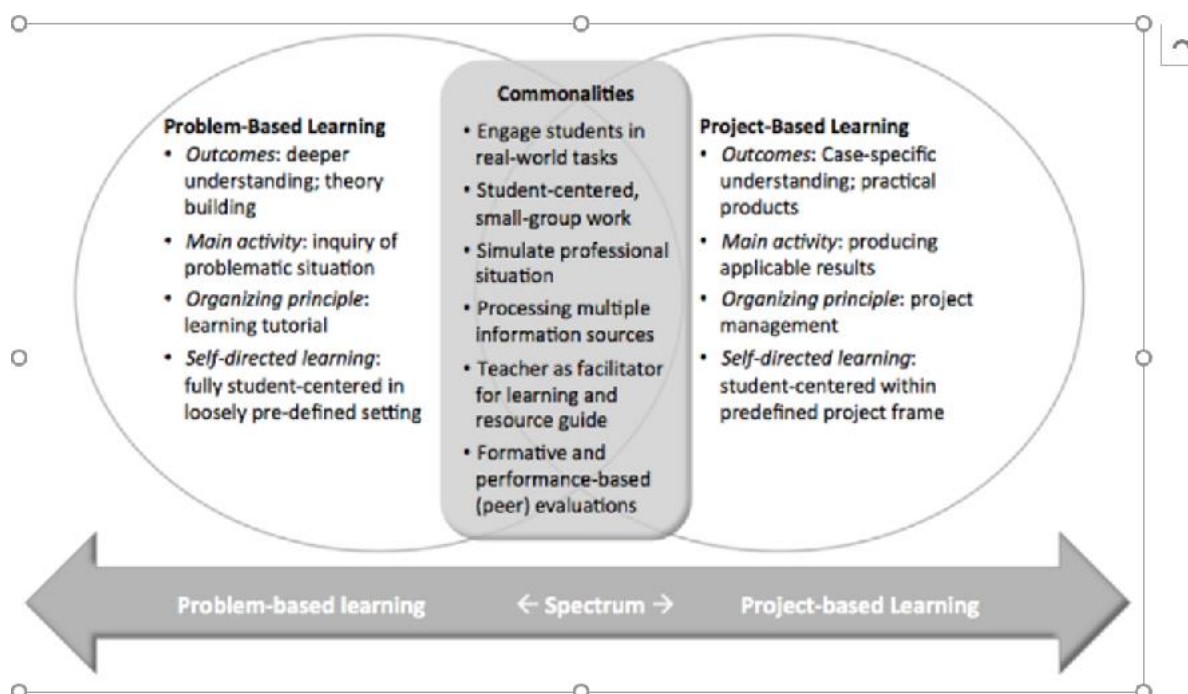
Gioco:

Abbiamo bisogno di divertimento durante il nostro processo di apprendimento. Un ambiente semplice, facce felici permettono alle persone di imparare meglio e più velocemente.

Progetto basato sull'apprendimento

Incoraggiare gli studenti creativi, PBL (Project-Based Learning) è un metodo basato sullo studio di un progetto che può coinvolgere tutte le discipline. Questo significa pianificare uno scenario reale per lo studente dove le idee possono essere connesse con la risoluzione di problemi reali. Grazie al metodo PBL, gli studenti sono più impegnati e possono migliorare la collaborazione, le competenze trasversali, favorire la creatività e rendere l'apprendimento divertente!

Da Wikipedia troviamo un'interessante immagine che ci permette di capire meglio che cosa è il PBL:



Flipped Classroom

Questo metodo deriva da metodi di apprendimento misto e prevede un cambiamento dei ruoli in classe, dove gli studenti devono studiare diverse materie e insegnare ai loro compagni di classe che cosa hanno imparato. In questo processo l'insegnante può cambiare i ruoli e diventare una sorta di coach, pianificatore degli studenti e spingerli ad esplorare con il loro gruppo di lavoro e a presentare nuove cose.

Sfida 1: Bacchetta Magica

Kit robotico: Lego Wedo 2 o Arduino o Microbit (in questo caso è sviluppato per Lego Wedo 2)

Software: Scratch 3.0 e Lego Digital Designer

Età: da 10 a 15 anni

Introduzione

Dobbiamo costruire una bacchetta con i pezzi del kit Lego WeDo 2, è molto semplice e non richiede manualità. Per questa ragione, da un punto di vista formativo, la lezione può essere organizzata iniziando dalla progettazione della bacchetta sia su fogli sia utilizzando il software gratuito Lego Digital Designer.

In questa fase gli studenti dovranno:

1. Disegnare la bacchetta sul foglio o sul computer (attraverso LDD)
2. Costruire la bacchetta con Lego WeDo con la sola accortezza di inserire il sensore di TILT nell'impugnatura della bacchetta.

Lo scopo dell'esercizio è creare attenzione e coinvolgimento da parte degli studenti, utilizzando tecniche narrative comuni (storie sui maghi). Grazie all'uso della bacchetta sarà semplice introdurre l'utilizzo del sensore di tilt e capire l'utilizzo del 'se' su scratch.

Storytelling

L'esercizio si presta a lavorare molto sulla parte narrativa, i bambini possono diventare parte di una storia, nella quale ci sono protagonisti dei maghi. Possono creare cappelli da mago e indossarli durante le lezioni. Si può lavorare sulle fiabe dedicate ai maghi utilizzando la costruzione e la programmazione di cui si parlerà , come fase della narrazione complessiva.

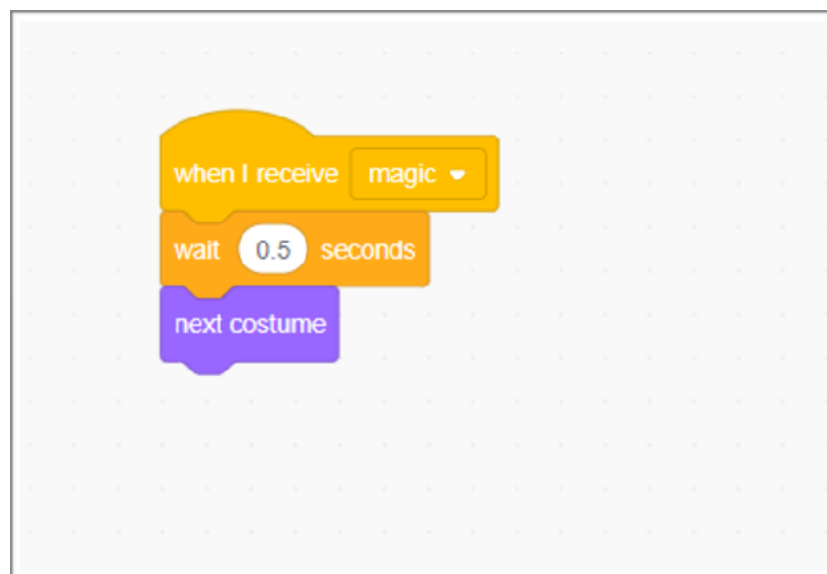
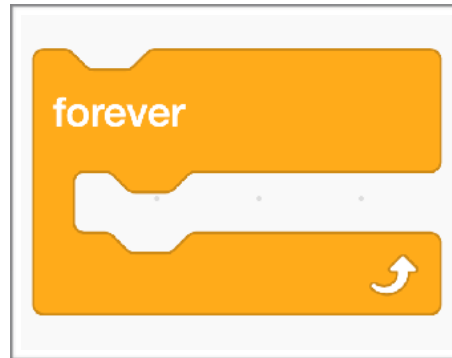
Primo programma

Il primo programma è utilizzato solamente per gestire la bacchetta e per cambiare l'aspetto dello sprite inserito in Scratch quando la bacchetta è mossa.

Per esempio, cambiando la posizione della bacchetta da verticale ad orizzontale

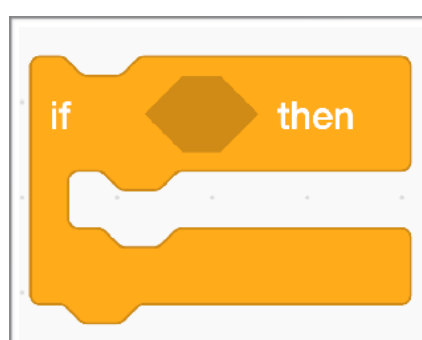
potrà cambiare la forma dello sprite.

Non sono richiesti prerequisiti particolari nel linguaggio di programmazione per far funzionare questo programma. In questo semplice esercizio, gli studenti possono trovare sul computer 2 importanti blocchi:

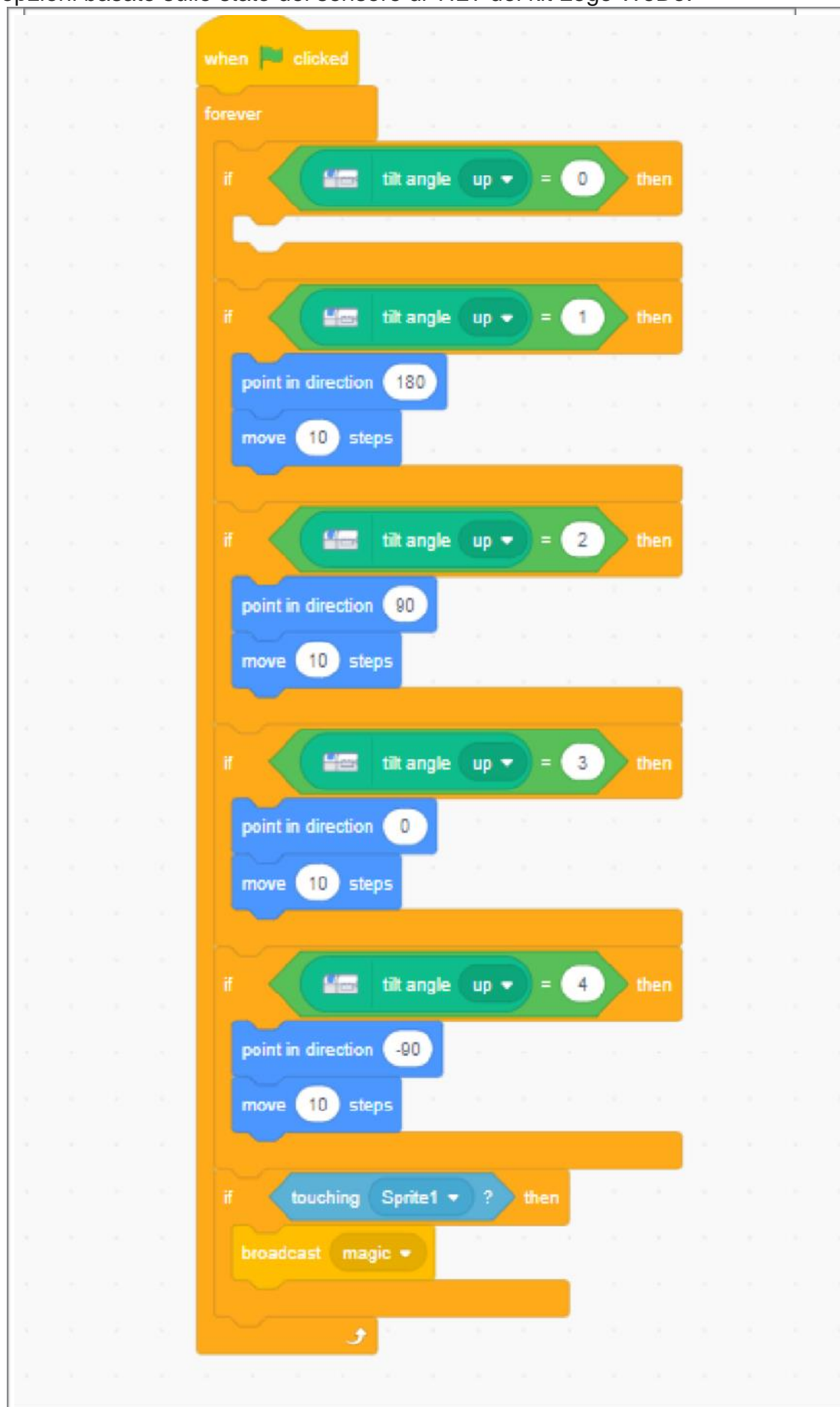


Questo blocco permette di creare un loop di ripetizione fino ad un numero infinito di volte. In questo caso, significa che possiamo controllare in ogni momento qual è la posizione della bacchetta.

Il secondo importante blocco è SE-ALLORA, attraverso il quale possiamo confrontare la posizione del sensore e se l'angolo è elevato possiamo cambiare l'immagine sullo schermo.

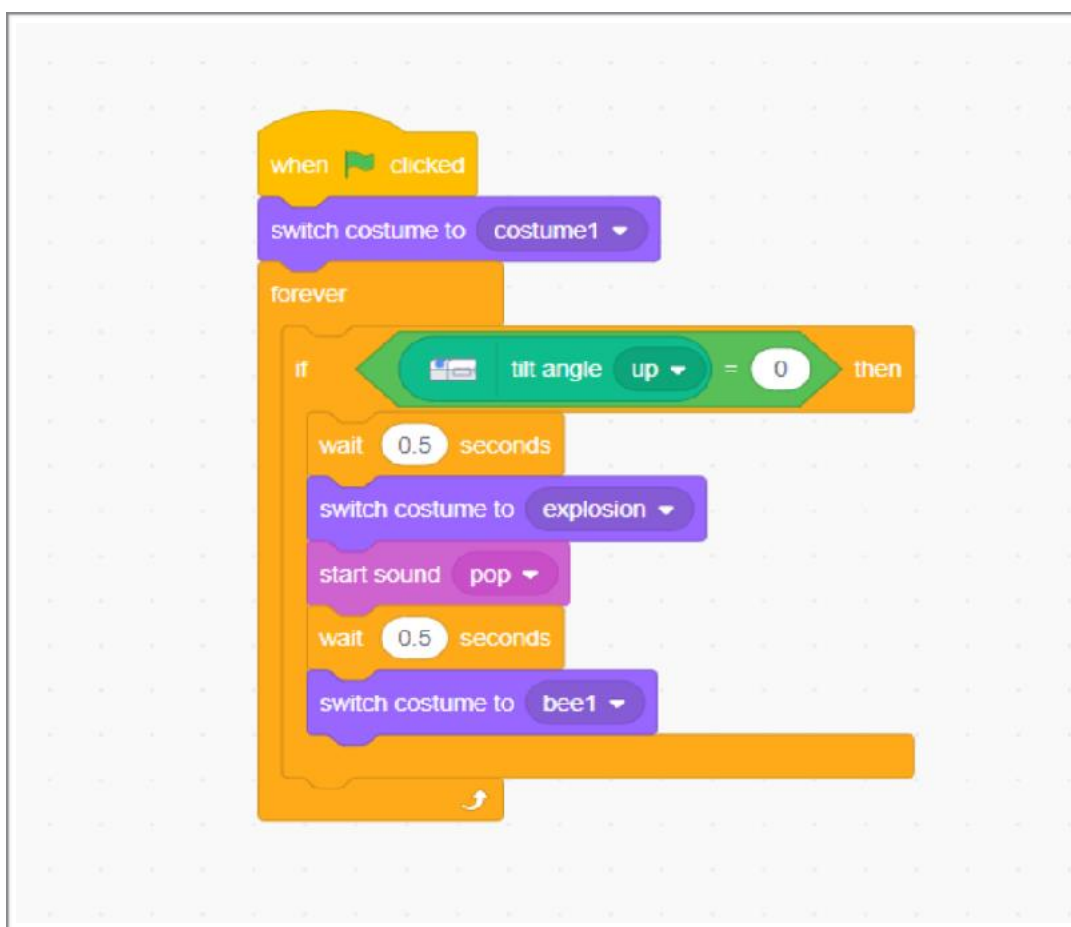


Grazie a questo semplice programma, gli studenti potranno capire le potenzialità del comando “SE” e costruire un semplice diagramma di flusso che include una o più opzioni basate sullo stato del sensore di TILT del kit Lego WeDo.



Programma 2

Il secondo programma non mostra nessuna estensione dell'orizzonte di insegnamento, ma solamente un consolidamento dei nuovi concetti di apprendimento. Il secondo esercizio prevede che al momento dell'esecuzione della magia è presente un suono e un effetto grafico che annuncia la magia. Dal punto di vista didattico non ci sono nuovi concetti, così gli insegnanti possono utilizzare questo esercizio per capire se gli studenti hanno capito i potenziali del software.



Inoltre questa parte di programma permette una forte personalizzazione da parte degli studenti (non solo il suono ma il design degli effetti magici). Questa personalizzazione permette un grande coinvolgimento degli studenti e un migliore e più immediato apprendimento.

Programma 3

Sui primi due esercizi si possono fare numerose varianti sull'argomento, di uguale difficoltà fra loro. Per aumentare il livello di difficoltà della programmazione, potete cambiare in questo modo la richiesta per gli student: "create una bacchetta magica che diventi un controllore di una stella nel video. Ogni cosa che viene toccata dalla bacchetta si trasforma."

Questa richiesta richiede la creazione di nuovi algoritmi:

1. Il movimento dello sprite secondo la posizione del sensore di TILT (4 stati identificati);
2. Collegamento di un flusso di programmazione a un altro (presente in un altro sprite) utilizzando il comando "Broadcast".

Sfida 2: Pesce e sensore di tilt

L'obiettivo è di mettere in gioco tutte le conoscenze acquisite fino a questo momento in un unico programma, che prevede l'interazione tra diversi sprite. La variabile utilizzata (già presente in Scratch) è chiamata "dimensione".

Questa parte è dedicata in particolare alla scuola secondaria di primo grado e al primo anno della scuola secondaria. La struttura del capitolo è leggermente differente dalle precedenti per accompagnare gli insegnanti alla programmazione in Scratch: non ci saranno esercizi graduati, ma solo un esercizio descritto step-by-step.

Scopo

L'obiettivo è creare un gioco nel quale un pesce predatore (solo digitale o anche "reale"), comandato attraverso il sensore di Tilt, possa mangiare solamente il pesce più piccolo. Ad ogni pesce mangiato il protagonista, il pesce predatore, cresce di dimensione. Il nemico è un pesce molto più grande, un super-predatore, che il pesce protagonista dovrà evitare fino a che non diventa abbastanza grande da non correre più rischi. Solo allora avremo raggiunto la fine del gioco.

Storytelling

Per costruire questo videogioco, dobbiamo introdurre alcuni elementi collegati alla trama di una storia. Abbiamo la necessità di introdurre la figura del protagonista, lo scopo del gioco (mangiare altri pesci) e la figura dell'antagonista, il nemico, che ostacola il raggiungimento dell'obiettivo.

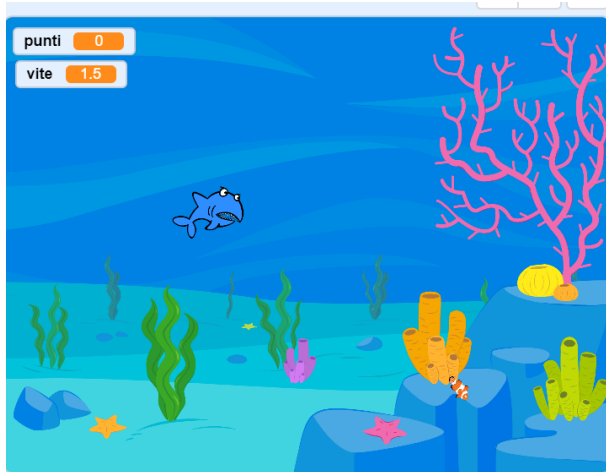
Questo può essere un primo step verso l'introduzione vera e propria della storia, in cui tutti gli attori sono chiaramente definiti: protagonista, aiutante del protagonista, antagonista, aiutante dell'antagonista, oggetto magico, obiettivo da raggiungere.

Il videogioco

Per poter realizzare il videogioco, viene presentata la programmazione di ogni elemento passo dopo passo e l'immagine del gioco finito.

Background (Stage)

Come nel capitolo 1, potete importare uno sfondo e modificarlo con l'editor del disegno, o se è possibile crearne uno nuovo, come è stato fatto con gli sprite. In questo esercizio non è necessario associare l'azione di sfondo, ma è possibile prevedere l'alternanza dei diversi sfondi; in questo caso devono essere mostrati in momenti opportuni, pianificandone l'aspetto.



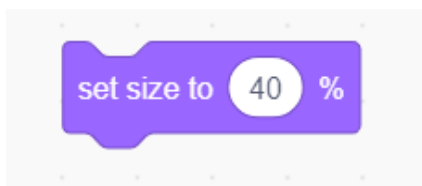
Sprite Protagonista - Movimento

Per controllare lo sprite del pesce protagonista basta inserire lo script nello script principale creato nell'esercizio del programma 3.

Sprite Protagonista – Inizializzazione variabile “dimensionale

Questo concetto alla base di ogni codice prevede l'inizializzazione delle variabili del programma. In questo caso la variabile che potremo utilizzare è già definita da Scratch (non dobbiamo crearla noi stessi) e si chiama dimensionale. L'inizializzazione viene fatta utilizzando il comando

"Port size to %" in the folder



Appearance (top left). We bring variability

"Size" at 30%.

Sprite piccoli pesci – movimento e iterazione

Nel videogioco ci saranno molti piccolo pesci, tutti con lo stesso script.

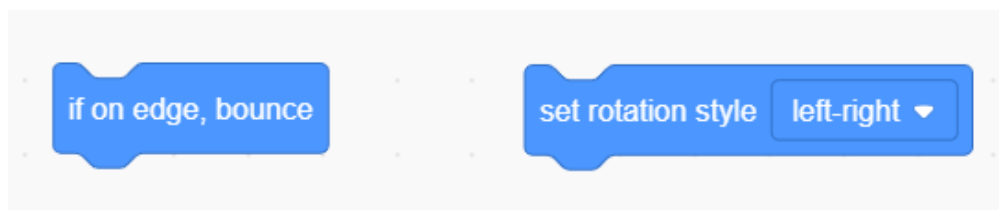
Fare il singolo sprite “pesce piccolo” e duplicarlo tutte le volte che è necessario.

Tutti gli sprite ne videogioco devono essere attivati attraverso lo stesso pulsante, in questo caso lo “Spazio” (queste sono azioni che vengono attivate “contemporaneamente”).

Quando si programma un videogioco si deve pensare che lo stato "mostra" o "nascondi" sia da inizializzare, soprattutto nel caso in cui durante il gioco si pensi di utilizzare anche il comando "Nascondi".

Per iniziare dovremo considerare di mostrare lo sprite e impostare la sua dimensione iniziale al 20%.

Si procede poi a programmare il movimento e l'interazione. Il movimento del pesce avviene lungo una linea ma bisogna avere la precauzione di inserire il comando "Rimbalza quando si tocca il bordo", per garantire la continuità del movimento. Se al momento del rimbalzo lo sprite si capovolge, è sufficiente selezionare il pulsante “gira solo a destra” e “a sinistra” (come mostrato nella seguente figura).

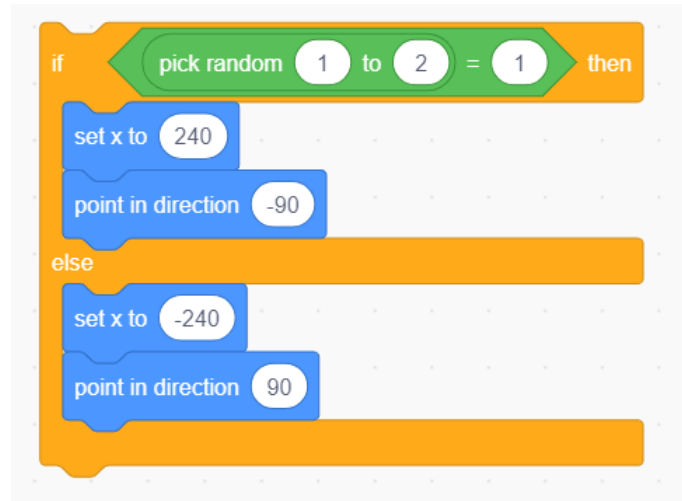


Sprite – Protagonista

Per programmare correttamente dobbiamo considerare che, nel momento in cui lo sprite del protagonista viene toccata da un pesce, bisogna valutare se il pesce è più grande o più piccolo. Se il protagonista incontra un pesce più piccolo (<30 nelle nostre impostazioni...), che si realizza con il comando “quando ricevo... dimensione xx”, sarà sufficiente suggerire al programma di aumentare la dimesione del protagonista di una percentuale a nostra scelta (nell’esempio 10).

Sprite – Pesce grande

Come nel caso dei pesci piccolo dovrà essere mostrato all'inizio. A differenza dei pesci piccolo si farà apparire il pesce grande in una posizione casuale (x,y), gestita dalla combinazione di numeri casuali che verranno inseriti con l'apposito script (come mostrato).



Casuale sarà anche la direzione di movimento del pesce. La dimensione iniziale del pesce grande viene impostata al 70%. Per garantire un movimento vario in tutto lo spazio disponibile, è possibile impostare i pesci in modo tale da rimbalzare su ogni bordo e avere una velocità casuale (il numero di steps è generato in modo casuale da 1 a 15). Come per i pesci piccoli, se il pesce grande tocca il protagonista deve inviare un messaggio nominato come dimensione (in questo caso 70).

Sprite Protagonista – ricevere il messaggio dal pesce grande

Quando il protagonista riceve il messaggio da pesce grande deve capire se sarà "mangiato" o se potrà affrontare l'avversario, condizione che sarà soddisfatta quando la sua taglia sarà almeno del 60%. Se la condizione è verificata, il videogioco termina con l'estremo ingrandimento del protagonista. Nel caso in cui il protagonista sia più piccolo del pesce superpredatore, il gioco terminerà con una scritta rappresentata da uno sprite che verrà attivate alla ricezione del messaggio "game over".

Sprite Game Over - Script

Game Over è uno sprite progettato dall'utente. La parola Game Over si attiva solamente quando riceve il messaggio Game Over, mentre all'inizio del programma è nascosto.

Sprite pesce Grande - Vittoria

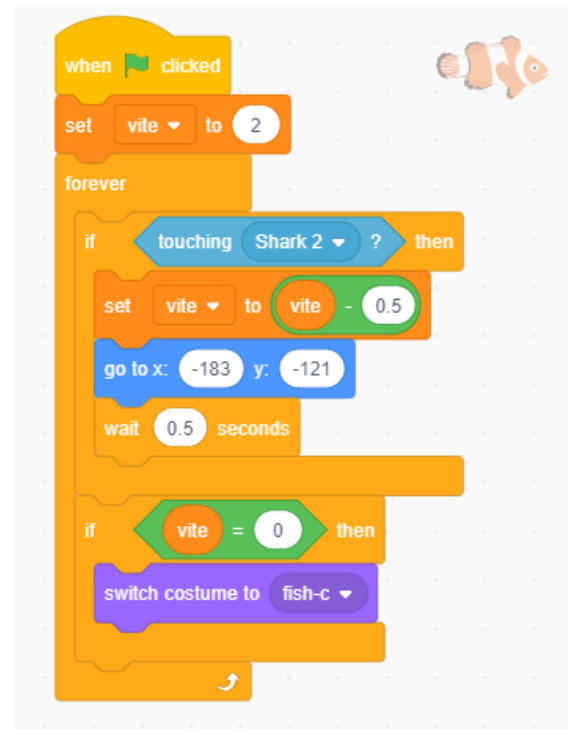
Il protagonista in caso di vittoria manda un messaggio "vinto" a tutti gli sprite. Quando il messaggio raggiunge lo sprite del Grande Pesce questo deve scomparire. Se lo si desidera, può essere aggiunto uno sprite "Hai vinto". Nel programma proposto il pesce Predatore diventa enorme e continua a nuotare.

Riassunto

La realizzazione di questo software è più complessa quindi per non appesantire la lettura non abbiamo inserito nel testo note didattiche come nelle precedenti dispense. L'insegnante nel condurre un laboratorio dedicato alla realizzazione di questo gioco dovrà chiarire alcuni concetti chiave.

Un esempio di script semplificato da cui partire potrebbe essere:





La variabile “dimensione”

Scratch permette di creare nuove variabili ma alcune sono già inserite e definite. "Dimensione" è il classico esempio di variabile locale, che può essere applicata solo allo sprite per il quale viene utilizzata, a differenza di altre variabili che possono essere utilizzate da più sprite.

L'uso di questa variabile non permette, ad esempio, di richiamarla negli script di altri sprite.

Grazie alla "dimensione" si può spiegare il concetto di variabile. In questo gioco è chiaro che il cambiamento della dimensione della variabile diminuisce il successo o meno del gioco.

Molto spesso quando le variabili vengono introdotte a scuola, si rischiano esempi postumi, creati ad hoc e non molto funzionali o troppo complessi. L'utilizzo in questo videogioco, tuttavia, fornisce un approccio semplice alla domanda variabili.

Numeri casuali

Per generare pesci che si muovono a velocità casuali è necessario utilizzare il generatore di numeri casuali di Scratch, questo ci permette di far capire agli studenti come questi "generatori" vengono utilizzati nel mondo dell'informatica.

Inviare - Ricevere

In questo programma, la comunicazione tra gli sprite è essenziale. Questo ci permette di introdurre flussi paralleli e di programmare flussi per realizzare complessi diagrammi di flusso preparatori.

Sfida 3: Oracle

Software: Scratch 3

Questa sfida è legata al mondo degli assistenti vocali che è una questione impotante sul mercato e tutte le più importanti aziende stanno producendo i loro assistenti vocali.

Per iniziare la lezione possiamo ispirare i nostri studenti a iniziare da un film di fantascienza, 2001 Odissea nello spazio e dopo con una panoramica di alcuni importanti assistenti vocali. Nel film di Kubric 2001 Odissea nello spazio possiamo trovare HAL2001 un assistente che può interagire con gli astronauti e controllare l'astronave. Dopo di che possiamo anche citare il mondo di Star Trek, dove tutto l'equipaggio può parlare con la "nave" chiedendo aiuto, suggerimenti e servizio.

Possiamo connetterci con il mondo reale presentando tutti (quasi tutti) i prodotti reali presenti sul mercato, ecco alcuni esempi:

Alexa (Amazon)



Google Home (Google)



Siri



Tutti questi assistenti vocali possono rispondere alle vostre domande controllando le informazioni su internet e utilizzando le applicazioni più importanti come la piattaforma musicale online, i servizi cloud, il mercato online.

Dal punto di vista etico possiamo citare con gli studenti questo caso studio legato all'uso della voce di un cantante famoso (come accade in Black Mirror). L'uso di un cantante famoso sta creando alcuni casi di un uso sbagliato dell'assistente vocale che sta sostituendo i veri amici degli adolescenti.

Da un punto di vista etico è importante citare questo caso studio e utilizzarlo con gli studenti per introdurre un uso etico della tecnologia.

L'introduzione dell'assistente vocale può essere utilizzata anche per introdurre concetti di AI.

Assistente vocale e Internet of things

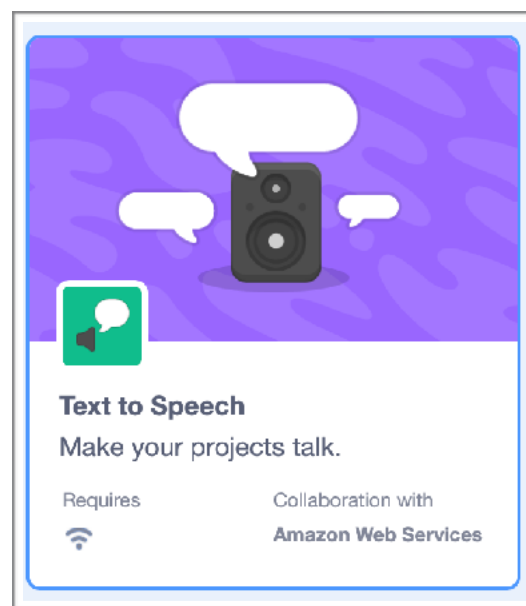
Presentiamo l'uso dell'assistente vocale da solo, ma in futuro tutti questi assistenti potrebbero essere la principale interfaccia con tutto ciò che è connesso ad internet nella nostra vita quotidiana.

Dopo questa introduzione possiamo scoprire come programmare un assistente vocale con Scratch 3:

In scratch non è possibile utilizzare il riconoscimento vocale, ma possiamo creare un programma in cui l'interazione sarà con tra la voce del computer e il testo dell'utente umano.

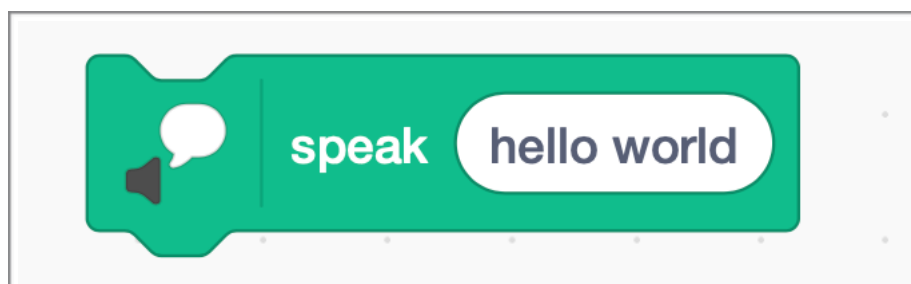
Per programmare un assistente vocale, una sorta di oracle, possiamo usare il comando "Text to speech".

Per utilizzare questa funzione, chiamata da Scratch Extension, dobbiamo cliccare in "Più estensioni" e possiamo selezionare questa immagine:



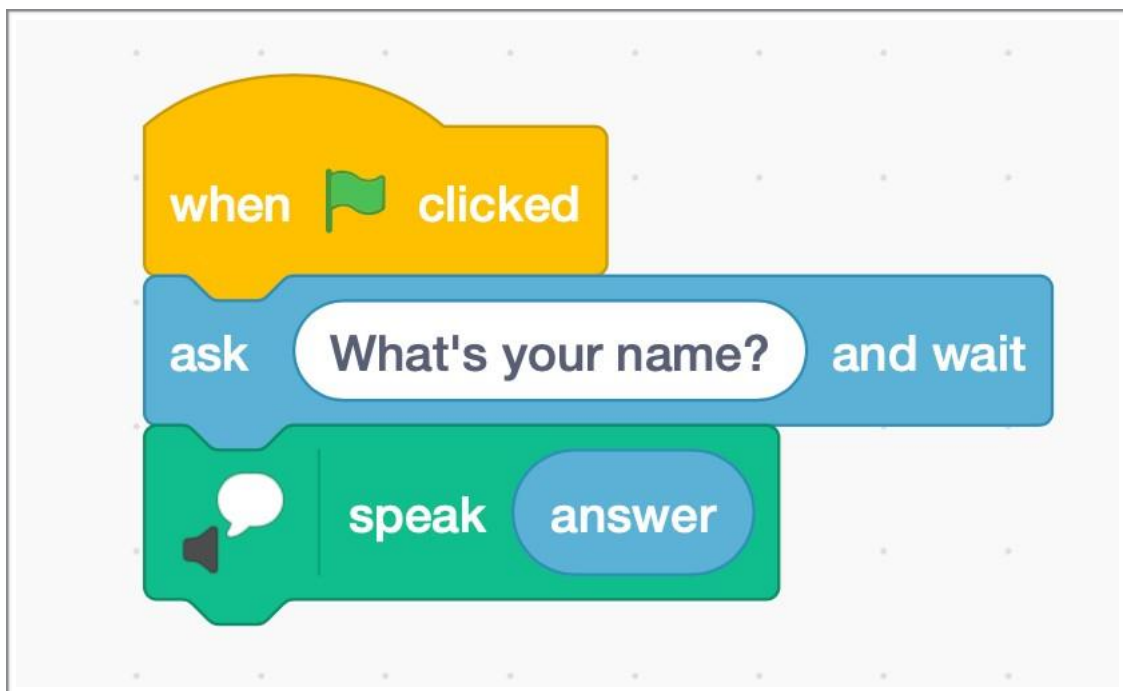
Che cosa è il text to speech?

Questo blocco permette al computer di leggere quello che scriviamo in formato testo, quindi l'interazione sarà interessante.



Possiamo usare questi blocchi

Grazie a questo programma possiamo chiedere il nome e scriverlo da tastiera. Utilizzando Text to speech il programma è in grado di leggere il nome. Possiamo migliorare il programma chiedendo alcune cose e ricevendo diverse risposte.



Focus su micro:bit



Scratch with micro:bit

Micro:bit è un microcontrollore tascabile progettato per aiutare i bambini ad imparare a programmare e creare con la tecnologia. Ha molte caratteristiche tra cui un display LED, pulsanti e un sensore di movimento. Può essere collegato e programmato con Scratch, MakeCode e Python.

Il BBC micro:bit contiene un accelerometro in grado di rilevare se è scosso o in che modo viene tenuto micro:bit. Oltre a rilevare se è capovolto, gli accelerometri possono rilevare alcune delle forze che agiscono su di esso. Micro:bit ha un

sensores di temperatura incorporato che può rilevare la temperatura attuale del dispositivo, in gradi e gradi Celsius. L'elenco completo delle caratteristiche è presentato nella tabella seguente:

Feature	Description
2 buttons	Programmable action push buttons
25 LED lights	Can be individually programmed to show shapes, text or numbers
USB connector	Connect to a computer for power or to load programs onto the <u>micro:bit</u>
Accelerometer	Senses if the <u>micro:bit</u> is being moved, tilted, shaken, or in free-fall, and at what acceleration
Compass	Detects which direction the <u>micro:bit</u> is facing
Processor	Where the program is executed
Radio	Communicate with other <u>micro:bits</u> for multiplayer games
Bluetooth antenna	Wirelessly sends and receives signals to Bluetooth enabled PCs, Smartphones, or Tablets
Reset button	Restarts the <u>micro:bit</u>
Battery socket	Power the <u>micro:bit</u> using batteries
Temperature sensor	Detects the current temperature of the <u>micro:bit</u> in degrees Celsius
Light sensor	The LEDs on the <u>micro:bit</u> can also act as a light sensor to detect ambient light
Edge Connector	25 external connectors, called Pins, on the edge of the <u>micro:bit</u> allow you to connect to other electronics hardware, including LEDs, motors, and other sensors. These can behave as inputs or outputs.

Per il momento, Scratch 3.0 non ha blocchi di codice per utilizzare la bussola, il termometro o il sensore di luce. Inoltre, solo 3 dei 25 pin sono utilizzati.

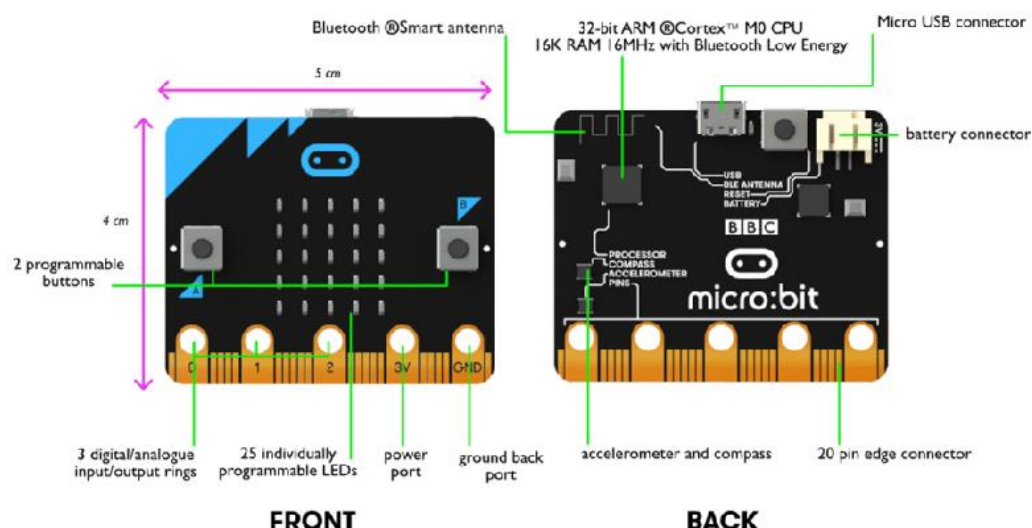


Image source: <https://www.pakronics.com.au/pages/microbit-in-australia>

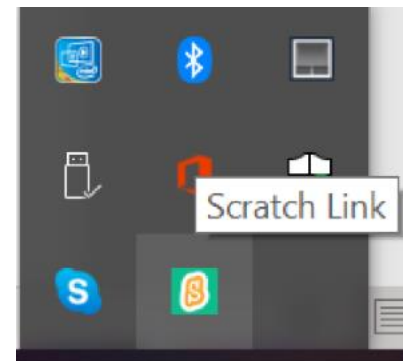
Come usarlo

Micro:bit può essere programmato sia su desktop (Mac, PC, Chromebooks, Linux, incluso Raspberry Pi) che su telefono. Per poter utilizzare micro:bit con Scratch è necessario seguire i passi successivi (per Windows).

Step 1: Installare Scratch Link da Microsoft store o utilizzare i link dalla pagina <https://scratch.mit.edu/microbit>

Step 2: Avviare Scratch Link e assicurarsi che sia in funzione. Dovrebbe apparire nella barra degli strumenti.

Step 3: Collegare micro:bit al vostro computer con un cavo USB. Micro:bit apparirà sul computer come un'unità chiamata 'MICROBIT'.



Step 4: Scaricare dal sito <https://scratch.mit.edu/microbit> il file Scratch micro:bit HEX

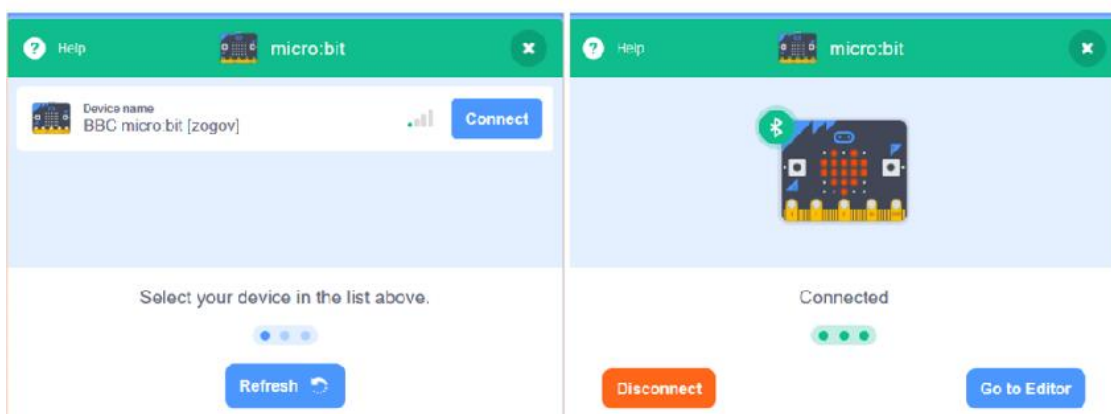
Step 5: Decomprimere l'archivio e trascinare il file HEX sul vostro micro:bit

Step 6: Alimentare Micro:bit con USB o una batteria.

Step 7: Utilizzare l'Editor di Scratch

Step 8: Aggiungere l'estensione micro:bit

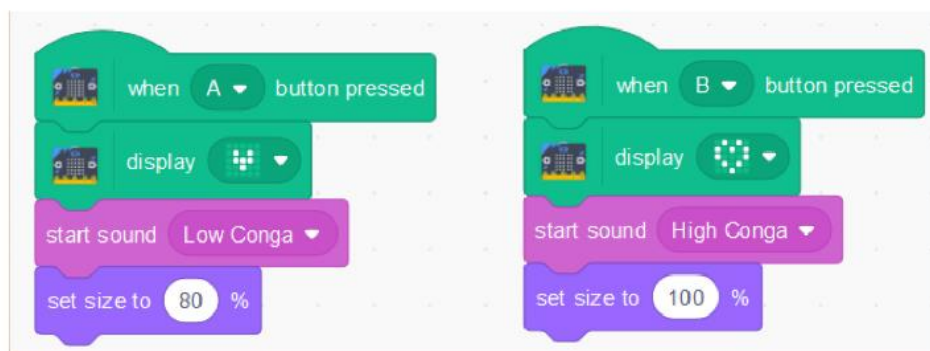
Step 9: Premere Connect e andare sull'Editor



Sfida di progettazione

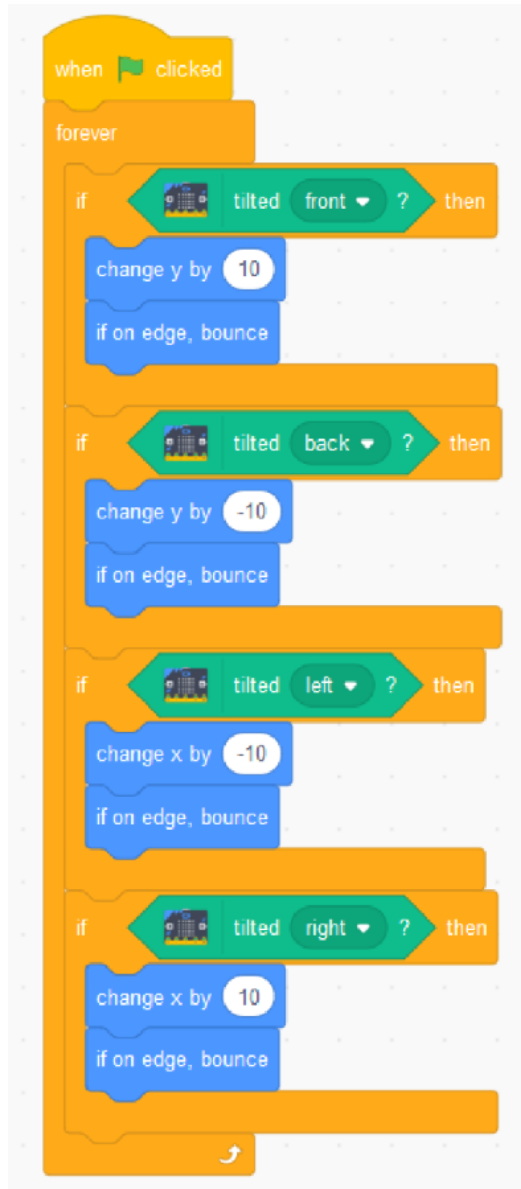
Sfida di progettazione 1: Un battito di cuore

L'applicazione (da <https://scratch.mit.edu/microbit>) è controllata dai due pulsanti di micro:bit. A seconda del pulsante premuto vengono visualizzati su micro:bit due cuori (uno più piccolo e uno più grande) e, allo stesso tempo, in Scratch un cuore cambia la dimensioni e vengono riprodotti due suoni diversi.



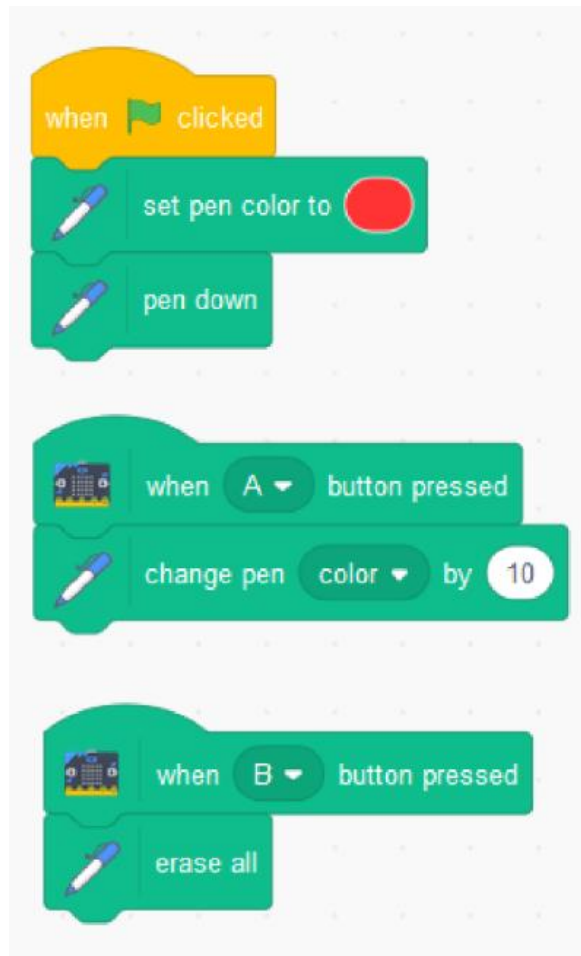
Sfida di progettazione 2: palla

Cambiando l'inclinazione del Micro:bit una palla può essere spostata sullo schermo.

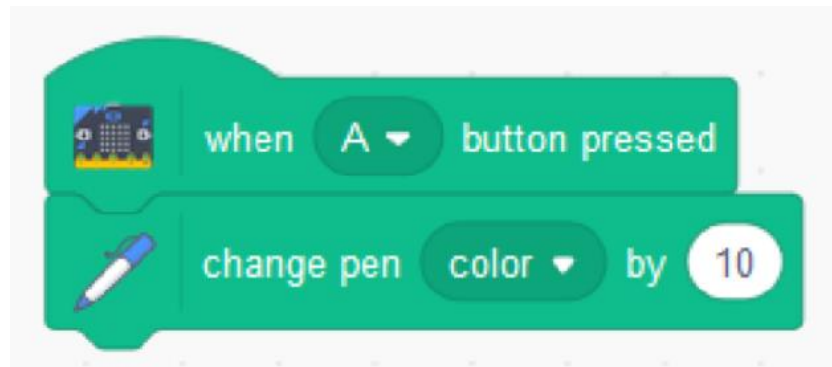


Sfida di progettazione 3: Una penna

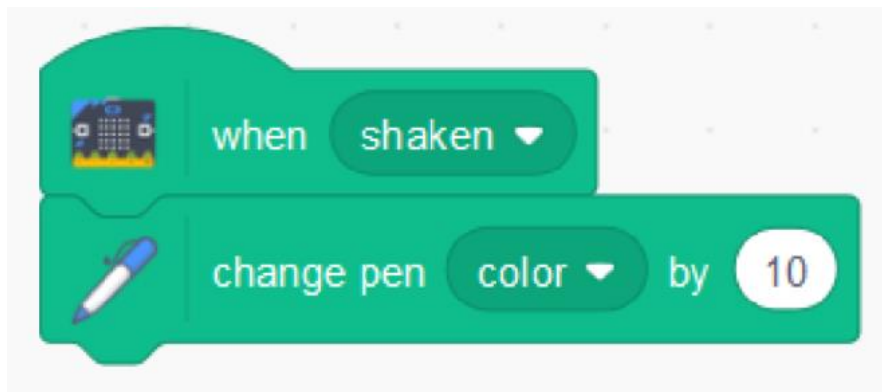
Aggiungere l'estensione della penna a Scratch. All'applicazione precedente, aggiungere gli script seguenti. Testare il risultato.



Se si sostituisce questo script



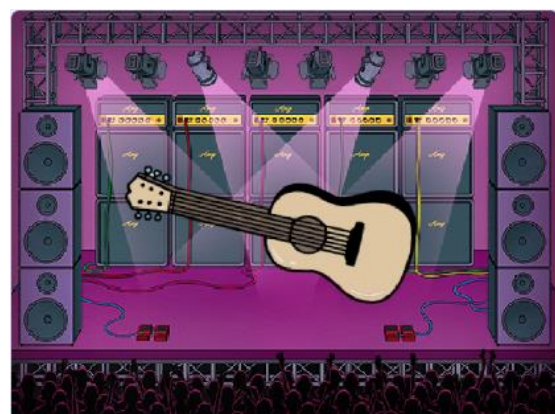
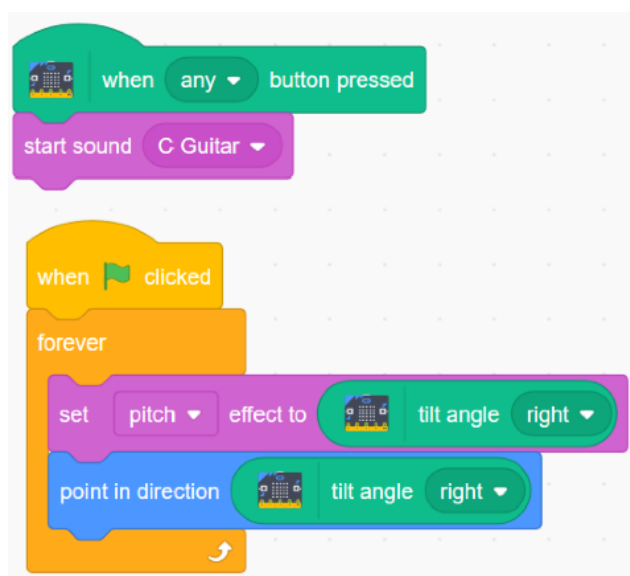
Con questo script



Allora il colore cambierà quando micro:bit verrà scosso.

Sfida di progettazione 4: Suona una chitarra

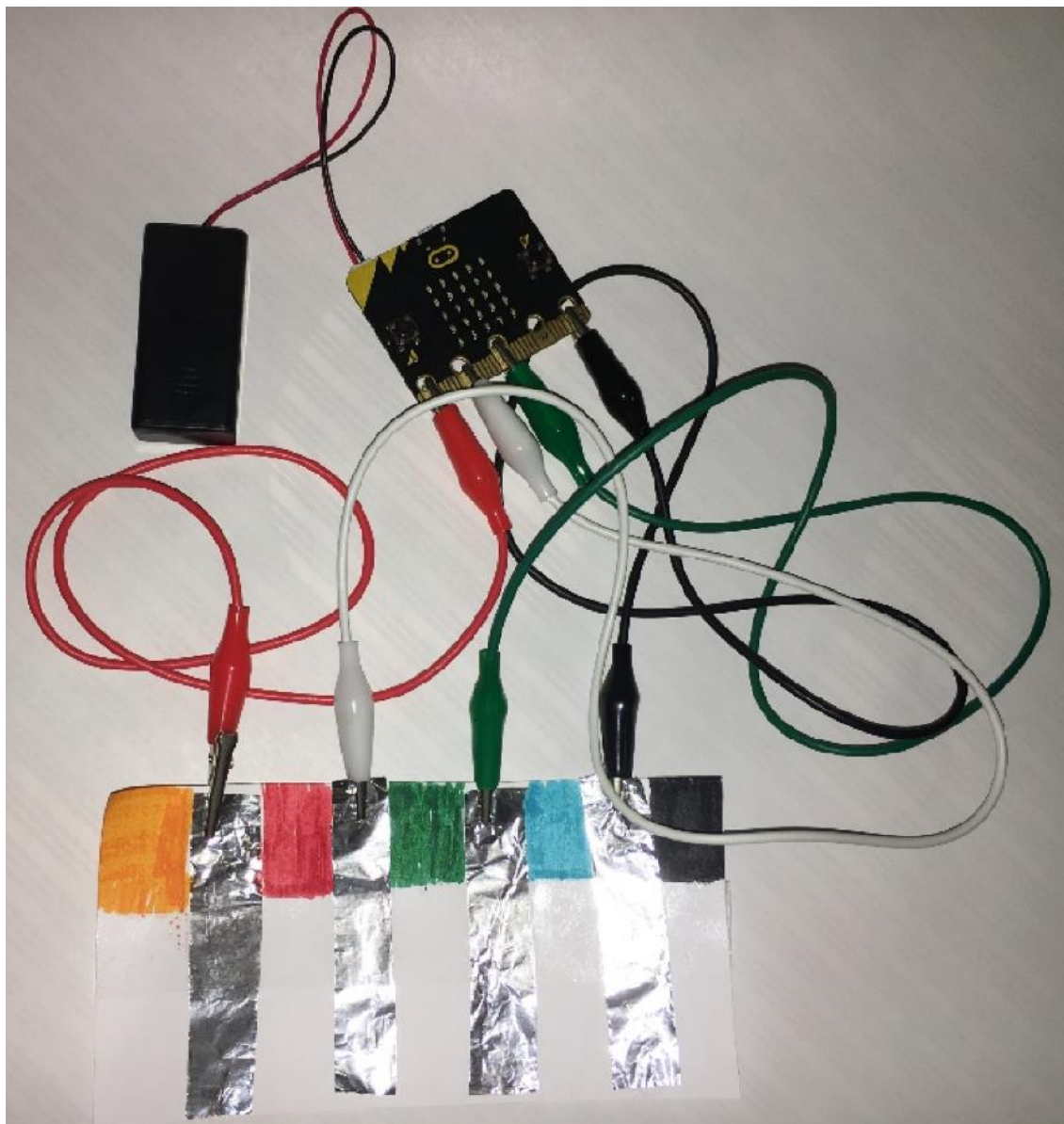
L'applicazione emette un suono quando si preme un qualsiasi pulsante micro:bit con un effetto che dipende dall'inclinazione del micro:bit.



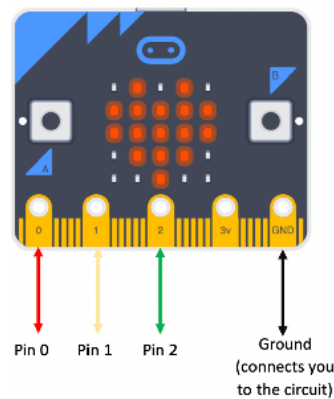
Sfida di progettazione 5: Suonare un pianoforte fai da te

Per costruire la prossima applicazione (ispirata all'esempio <https://microbit.org/scratch/>) avrete bisogno della schedina micro:bit , 4 cavi con clip a coccodrillo, stagnola, forbice, colla e carta.

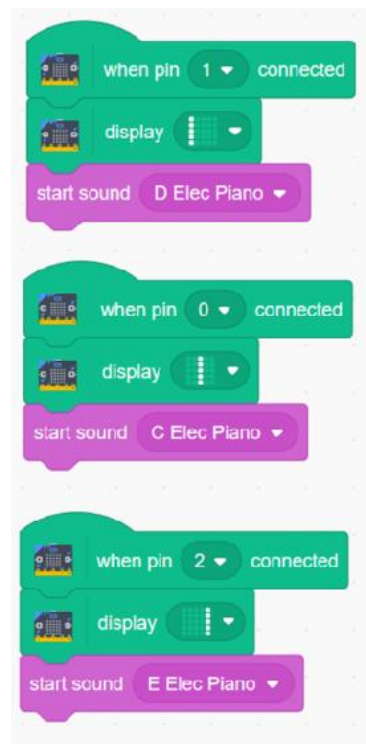
Costruisci un pianoforte come quello nell'immagine seguente. Collegare ogni pin da 0 a 2 e la terra ad una banda di stagnola.



Lo schema di questo pianoforte è nell'immagine seguente:



Scrivi gli scripts:



Testate l'applicazione toccando il pianoforte (bande di stagnola) (un dito deve toccare la banda di stagnola collegata a terra per chiudere il circuito).

Altre sfide da fare

1. Costruisci il gioco “carta, forbice, sasso”. L’applicazione dovrebbe far visualizzare casualmente una carta, forbice, sasso quando si scuote il micro:bit.
2. Creare il proprio dispositivo indossabile con un'interfaccia in Scratch. Progettare un case/support per il vostro dispositivo e stamparlo con la stampante 3D.

Risorse

<https://microbit.org/scratch/>

<https://llk.github.io/microbit-extension/iste18/>

<https://make.techwillsaveus.com/>

<http://libraryadventuring.blogspot.com/2018/10/coding-and-making-with-bbc-microbit.html>

<http://blog.sparkfuneducation.com/five-wearable-projects-with-microbit>

<https://scratch.mit.edu/discuss/youtube/44Xo76Bbqil/>

Riferimenti bibliografici

<https://microbit.org/scratch/>

https://diyodemag.com/education/kids_coding_scratch_30_meets_micro_bit

MakeCode per micro:bit

Microsoft MakeCode è un ambiente web-based per imparare a programmare con dispositivi computazionali fisici come micro:bit. MakeCode è gratuito e funziona su tutte le piattaforme e i browser.

MakeCode è disponibile come editor online per:

-micro:bit - <https://makecode.microbit.org/>

-Circuit Playground Express - <https://makecode.adafruit.com/>

-Minecraft - <https://minecraft.makecode.com/>

-LEGO® MINDSTORMS® Education EV3 - <https://makecode.mindstorms.com/>

-Arcade - <https://arcade.makecode.com/>

-Chibi Chip - <https://makecode.chibitronics.com/> e come applicazione per:

-micro:bit

-Adafruit

-Cue by Wonder Workshop



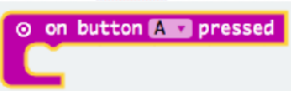
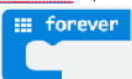



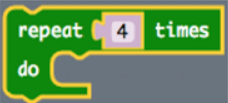
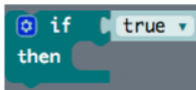



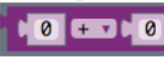
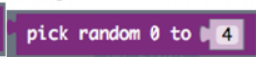
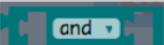

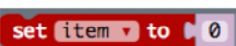
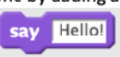
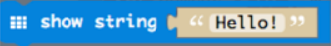

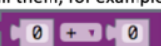
L'applicazione MakeCode per micro:bit ha alcune caratteristiche extra rispetto all'editor online. Con l'applicazione desktop il micro:bit può essere programmato direttamente via USB, senza bisogno di trascinare il file sul micro:bit e leggere direttamente i dati seriali da micro:bit.

Scratch vs MakeCode per micro:bit

Nella pagina di supporto micro:bit

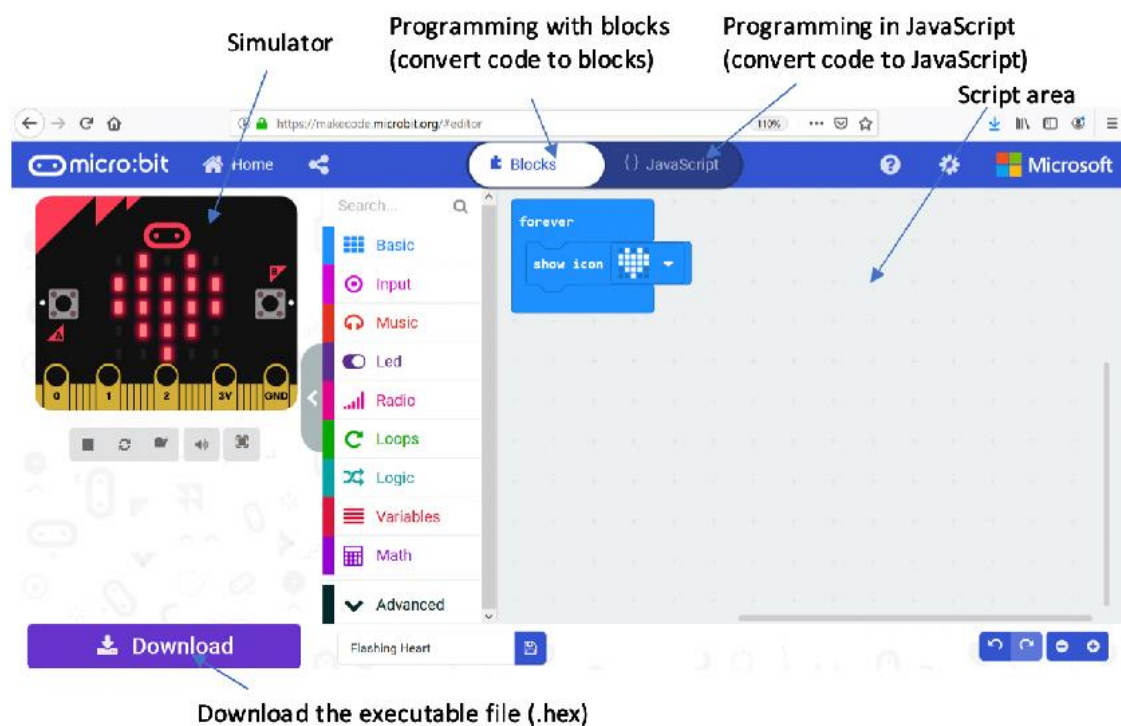
<https://support.microbit.org/support/solutions/articles/19000080171-movingbetween-scratch-and-makecode>

viene pubblicata una tabella di confronto, rappresentata di seguito, tra Scratch e Makecode per micro:bit, al fine di enfatizzare le sottili differenze tra diversi blocchi importanti. La tabella potrebbe essere utile per l'utente di uno dei due editor online, appassionato di micro:bit, per iniziare ad utilizzare l'altro.

Scratch	Makecode
Events wait for a user action, like clicking the green flag in scratch or pressing a button on the keyboard  	Input waits for a user input like pressing the A button or shaking the micro:bit 
Control is about the flow of your program tasks. In scratch you can add a forever , repeat or if block to an event to trigger it.	Makecode breaks controls into programming concepts. Forever is its own loop (it's triggered as soon as the micro:bit is powered on). 
  	Repeats are found in the Loops menu of Javascript Blocks  And if blocks are found in the Logic menu 
Operators let you do arithmetic and make comparisons   	Arithmetic and random number pickers can be found in Math , whereas a comparison between something and something else is found in Logic   
Data lets you define variables that might change within your program. Here we have made a variable called item 	We can define these in the Variables menu of makecode 
Looks let you display actions on the screen, which can be done by adding a say block. The word in the box is called a string 	To show a word on the micro:bit display we can use show string , found in the Basic menu 
Blocks that let you edit them have white backgrounds that you can type in for example operators have circular input areas 	Blocks that you can edit usually look like jigsaw pieces and may already have an example in them, for example Math sums have a '0' in the block 

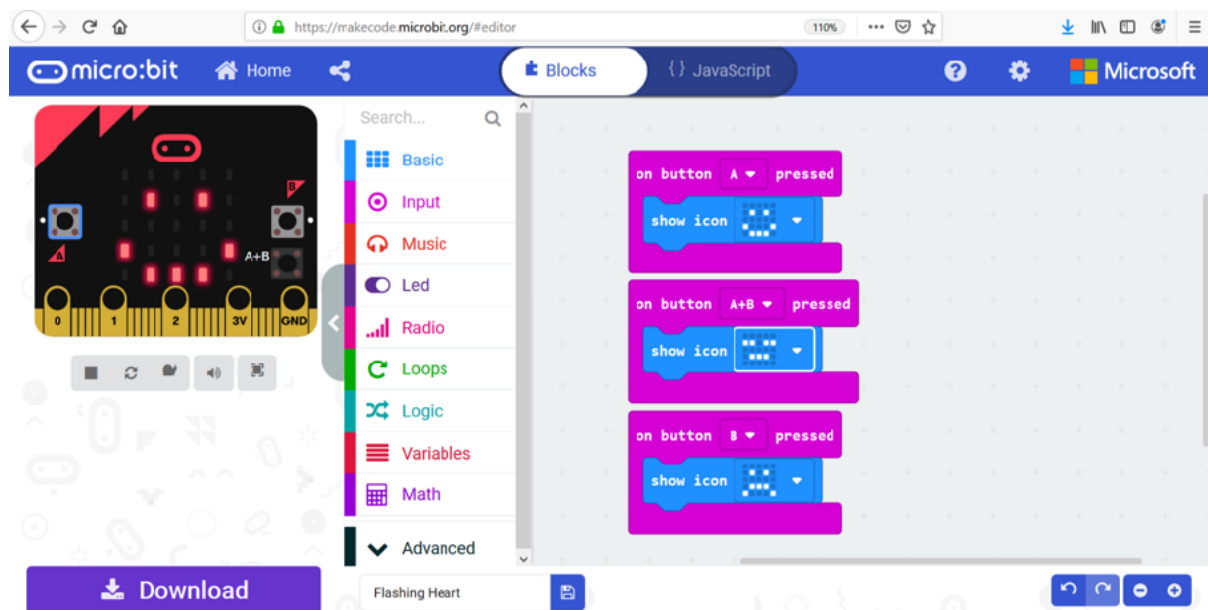
Un grande vantaggio di Scratch, rispetto al MakeCode online, è che permette la programmazione OTA (Over-the-Air) tramite comunicazione Bluetooth. Il grande svantaggio è che Scratch fornisce un set di blocchi base per micro:bit, solo 10. Per questo momento, il set più ricco di blocchi micro:bit in MakeCode lo rende preferibile.

Interfaccia



Sfida 1: Volti

Questa applicazione mostra una faccina sorridente quando si preme il pulsante A, una faccia triste quando si preme il pulsante B e una faccia addormentata quando si premono entrambi i pulsanti A e B.



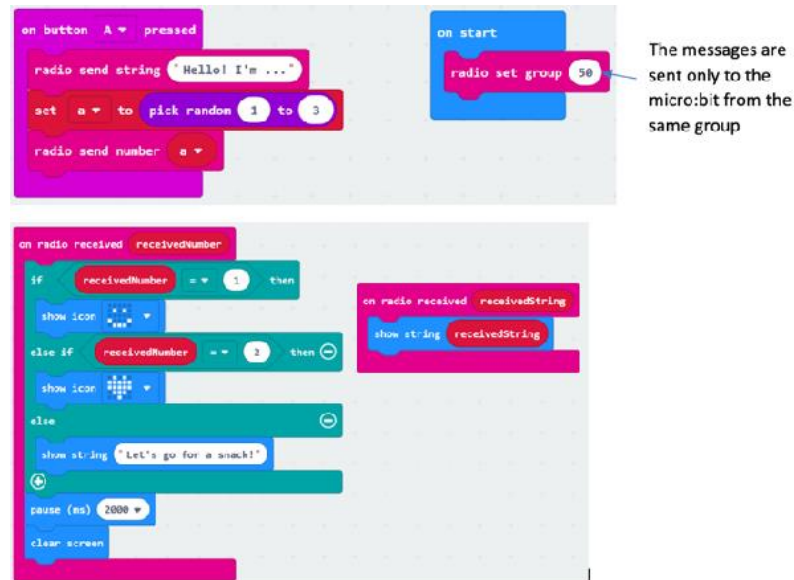
Nell'area di simulazione ci sono i pulsanti A, B e A+B che possono essere usati per simulare cosa succede quando si premono i pulsanti micro:bit.

Per mettere l'applicazione sulla schedina micro:bit è necessario scaricare lo script (come un file .hex). Collegare il micro:bit al computer con un cavo USB. Individuare il file .hex scaricato e trascinarlo sull'unità MICROBIT.

Questa applicazione si ispira ai tutorial di <https://makecode.microbit.org/projects/>. Provate altri tutorial!

Sfida 2: Chat

L'applicazione successiva permette a due o più schede micro:bit di comunicare tra loro via radio. Le variabili *receivedString* e *receivedNumber* vengono trascinate fuori dal blocco *on radio received*. L'applicazione deve essere caricata su ogni micro:bit.



Esempi con connessione radio possono essere trovati su:

<https://makecode.microbit.org/projects/radio-games>

e su:

<https://www.instructables.com/id/Radio-Signals-on-Microbit/>.

Sfida 3: musica e luci

L'applicazione successiva permette di fare musica variando l'intensità della luce sui sensori di luce micro:bit.

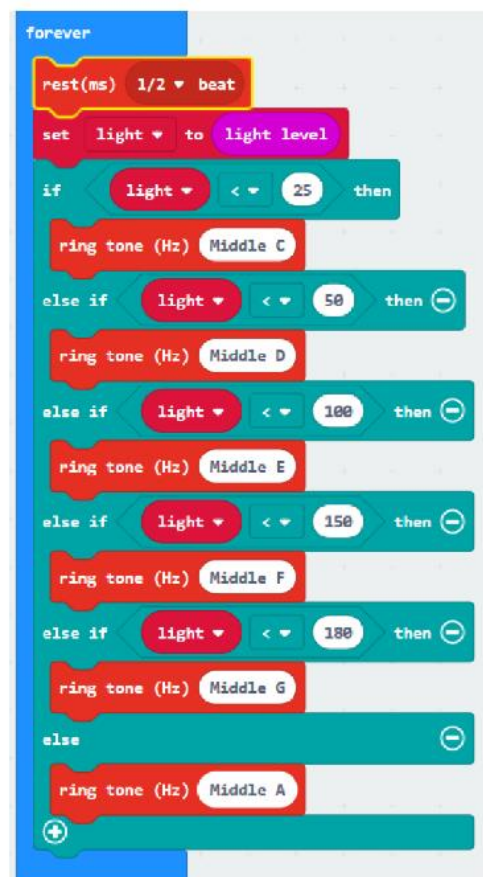
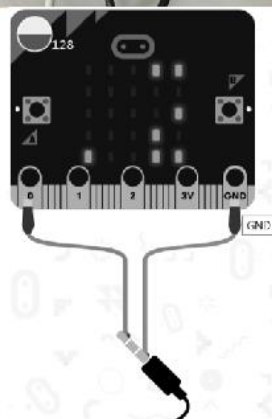
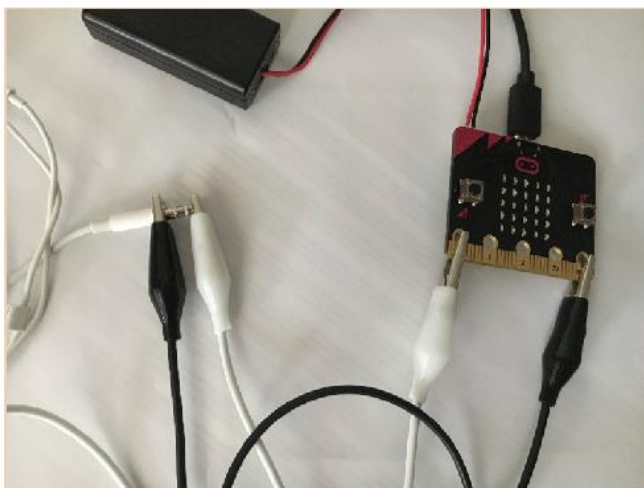
Materiali richiesti:

1x Micro:bit

1x Headphones

2 cavi con clip a coccodrillo 1x cavo USB

Per cominciare, dovrete collegare le cuffie al micro:bit come nelle seguenti immagini:



Definire una nuova variabile chiamata *light* e scrivere il codice. Copiare il file .hex sul drive micro:bit. Il livello di luce è un valore compreso tra 0 (scuro) e 255 (chiaro). La luce viene misurata utilizzando vari LED dello schermo micro:bit.

Sfida 4: Musica e... frutta e verdura

Si tratta di un'applicazione divertente che utilizza frutta e verdura per chiudere il circuito e fare musica.

Materiale necessario:

1x micro:bit

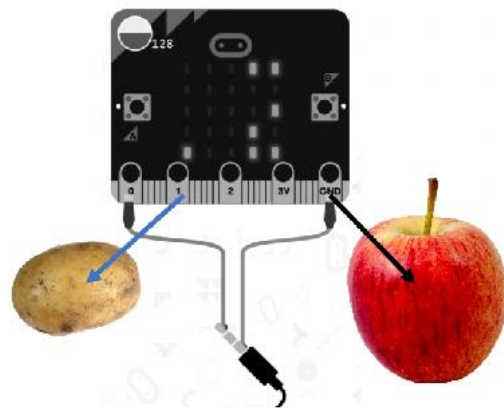
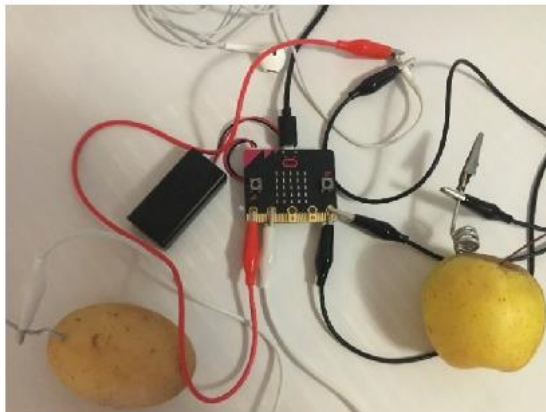
1x cuffie

4x cavi con clip a coccodrillo

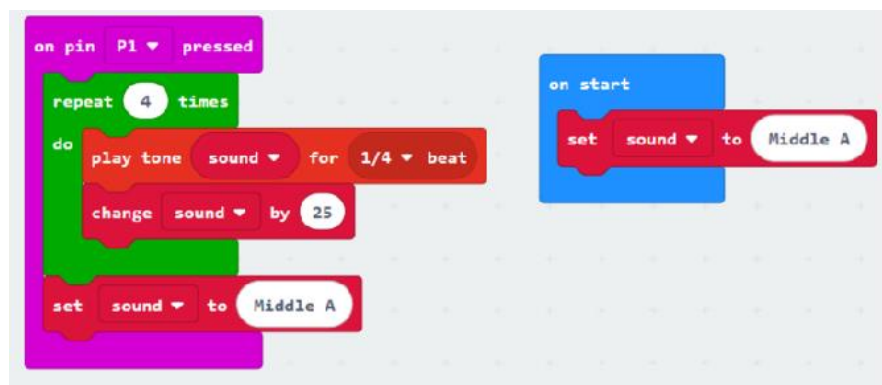
1x cavo USB

2x frutta e verdura (patate, banane, arance, mele, etc...)

Effettuare i seguenti collegamenti:



Creare una variabile denominata suono per memorizzare una nota musicale. Scrivere il codice successivo.



Salvare il file .hex e copiarlo sul drive micro:bit. Creare musica tenendo la frutta/verdura collegata a terra (la mela nel nostro esempio) e toccando l'altra frutta/verdura (collegata al pin 1).

Potete collegare un altro frutto/verdura al pin 2 per creare altri suoni. In questo caso potete duplicare il codice del pin 1, selezionare il pin 2 e modificare, ad esempio il valore 25 con -25.

Sfida 5: Servomotore

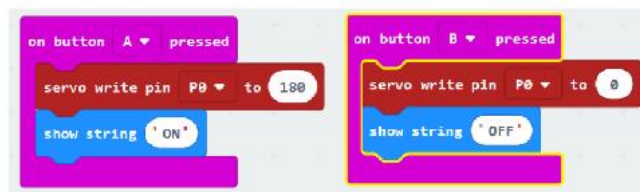
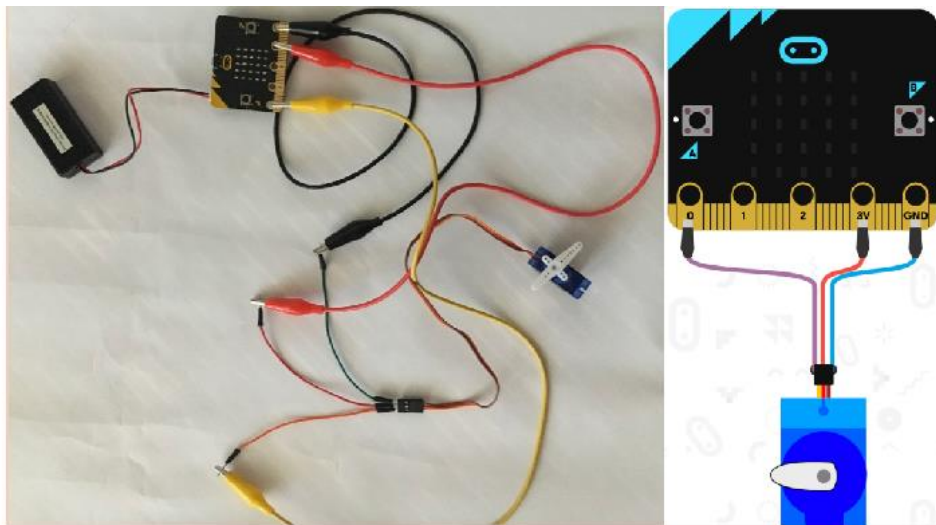
Questa applicazione mostra come collegare un servomotore al micro:bit.

Materiale necessario:

- 1x micro:bit
- 3x cavi con clip a coccodrillo
- 1x cavo USB
- 3x filo maschio- maschio
- 1x servomotore TowerPro SG90

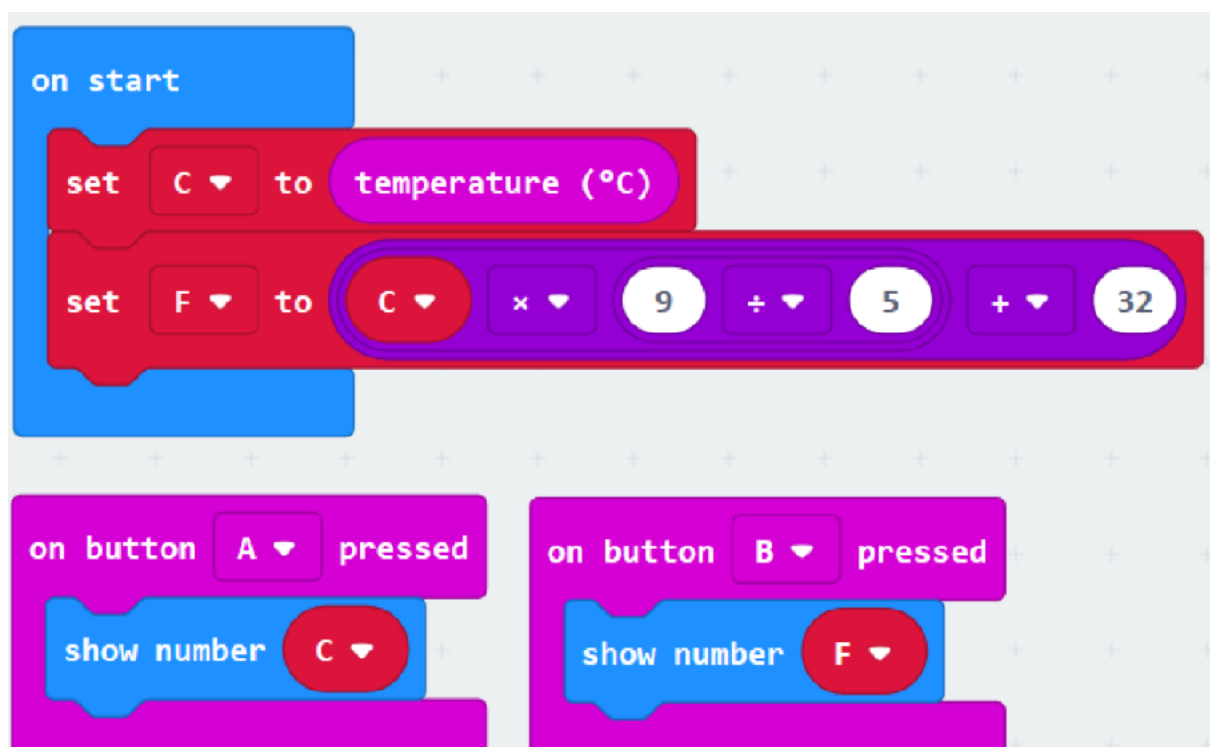
Il servomotore deve essere collegato come segue:

micro:bit	servomotor
GND	Brown wire
3V	Red wire
P0	Orange wire



Sfida 6: Termometro

Questa applicazione utilizza il sensore di temperatura incorporato per visualizzare la temperatura in gradi Celsius, quando si preme il pulsante A e in gradi Fahrenheit si preme il pulsante B.



Per ottenere un valore più vicino a quello reale, dovete confrontare il valore del micro:bit con un valore di un termometro reale. Poi il programma può essere modificato sottraendo la differenza dal numero che mostra il micro:bit.

Sfida 7: Bussola

La seguente applicazione mostra in quale direzione cardinale è posizionato il micro:bit. Dopo aver copiato il file .hex sul micro:bit, il chip chiederà la calibrazione. Per questo si dovrà inclinare il micro:bit in tutte le direzioni fino a quando tutti i LED sono accesi. Saprete che la calibrazione è riuscita quando viene visualizzata una faccia felice.



Sfide aggiuntive

1. Costruire un contapassi.
2. Costruite la vostra applicazione tenendo presente la materia che insegnate!

Risorse

<https://makecode.microbit.org/>

<https://makecode.com/labs>

<https://makecode.microbit.org/projects/>

<https://www.itpro.co.uk/desktop-hardware/26289/13-top-bbc-micro-bit-projects>

<https://www.101computing.net/category/bbc-microbit/>

<https://support.microbit.org/support/solutions/articles/19000080171-moving-between-scratch-and-makecode>

Riferimenti bibliografici

<https://makecode.microbit.org/projects/>

<https://support.microbit.org/support/solutions/articles/19000080171-moving-between-scratch-and-makecode>

“Whether you want to uncover the secrets of the universe, or you just want to pursue a career in the 21st century, basic computer programming is an essential skill to learn.”

- *Stephen Hawking, theoretical physicist and cosmologist*