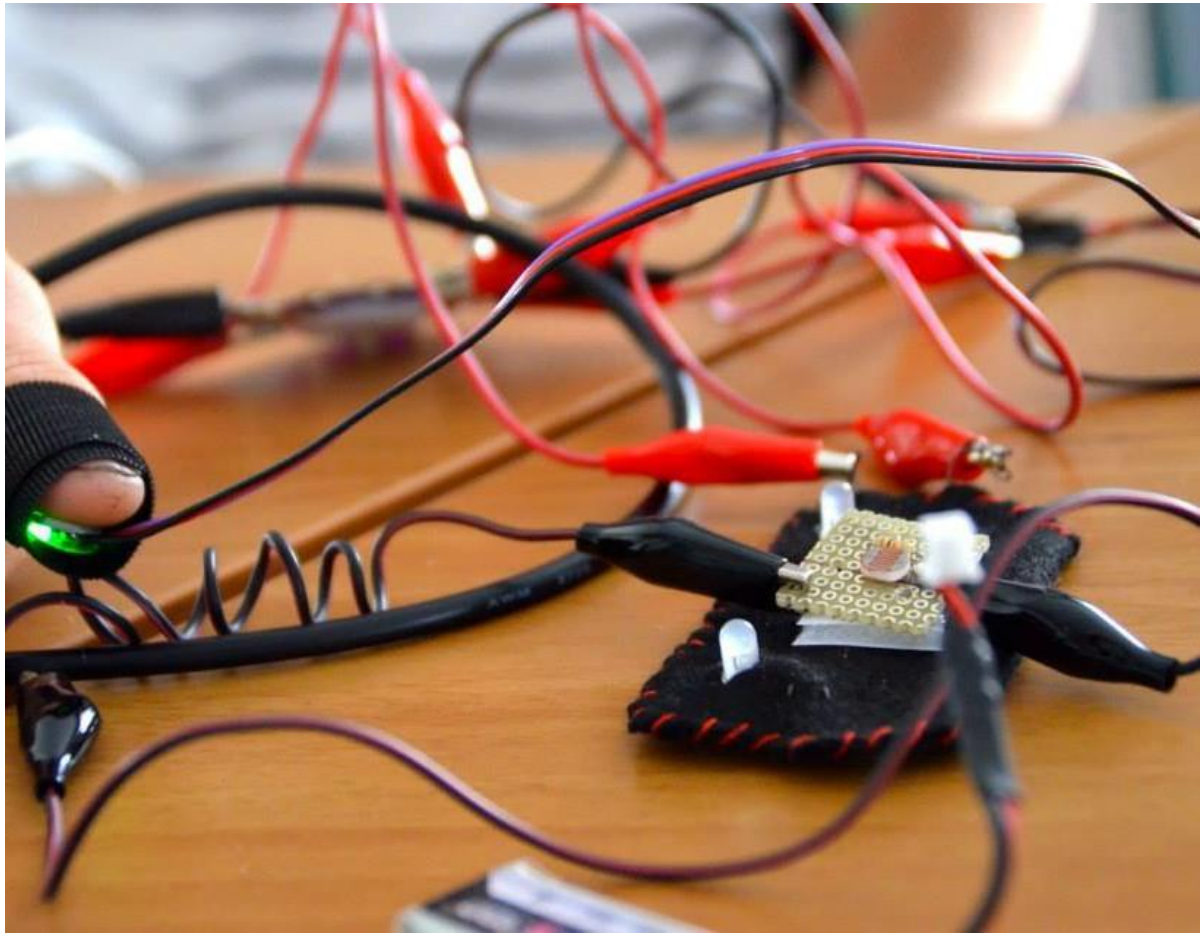


IOT-uri educaționale

Obiectivul acestui manual este de a prezenta diferite metodologii, scenarii de învățare, IoT-uri pentru educație și activități educaționale de codare.



Pentru a prospera, ei trebuie să învețe să proiecteze soluții inovatoare pentru problemele neașteptate care, fără îndoială, vor apărea în viața lor. Succesul și satisfacția lor se vor baza pe capacitatea lor de a gândi și de a acționa creativ. Doar cunoașterea nu este suficientă: ei trebuie să învețe cum să-și folosească cunoștințele în mod creativ.

- Mitchel Resnick, MIT Media Lab

O introducere în Internetul Lucrurilor și a dispozitivelor portabile în educație

În cadrul acestui manual puteți învăța cum să folosiți o parte dintre editoarele online simple pentru a programa și interacționa cu senzori și ieșiri simple. Internetul Lucrurilor (en. „*Internet of Things*”) este un domeniu în creștere pe piață, ce include de la termostate la ceasuri inteligente.

În domeniul educației va fi importantă introducerea acestor tehnologii, prin aplicarea teoriei 4P, deoarece pentru elevi aceste activități sunt antrenante.

Proiect, Lucru în pereche, Joacă și Pasiune (en. Project, Peer, Play și Passion) este o metodologie introdusă de Mitchel Resnick de la MIT Lifelong Learning Lab (MediaLab) și este foarte potrivită pentru activitățile educaționale IoT. Mai mult decât atât, în acest manual este prezentată posibilitatea de a crea programe pentru diferite dispozitive programabile, cu editoare asemănătoare cu Scratch 3.

În scenariile de învățare se folosesc și programe precum Snap for Arduino sau Makecode. Toate mediile de programare sunt gratuite și compatibile cu majoritatea platformelor robotice importante, cum ar fi Lego, Microbit, Arduino și Raspberry PI.

Introducere în Arduino și platforme compatibile Arduino

Am ales să prezentăm doar placa Arduino și platformele compatibile deoarece acestea sunt open source și în cadrul proiectelor UE considerăm etică utilizarea platformelor ieftine și open source precum Arduino și Elegoo.

Aceste plăci sunt complet compatibile cu o multitudine de senzori și actuatore.

Arduino este cea mai populară platformă electronică open source care schimbă lumea educației, datorită prețului redus și ușurinței în programarea acestora și crearea de prototipuri.

O placă programabilă uzuală e compusă dintr-un micro-controler pe 8 biți cu chipuri diferite din familia Mega AVR. Fiecare placă are intrări și ieșiri analogice și digitale.

Software online gratuit

În cadrul acestui proiect alegem să folosim doar software gratuit pentru a programa obiecte. Cel mai popular program este Scratch, acum aflat la versiunea Scratch 3.

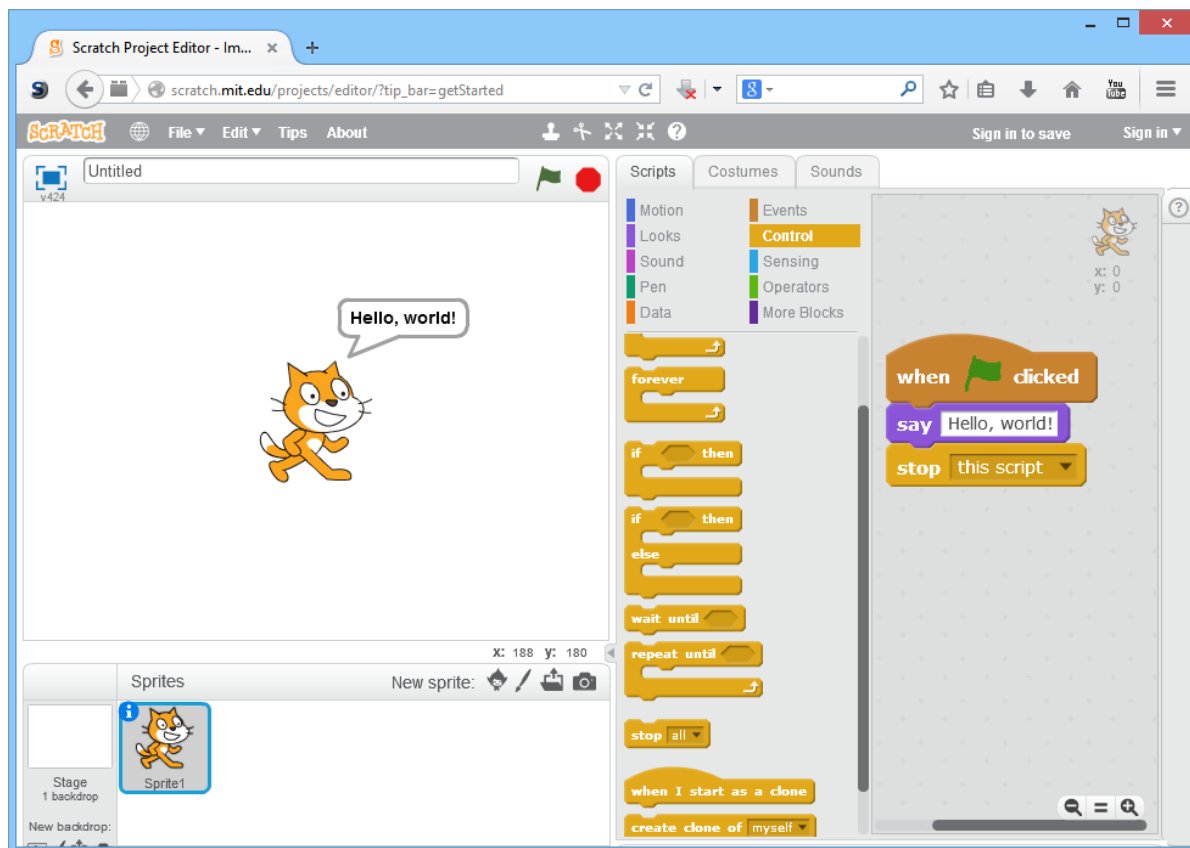
Scratch

Scratch este un editor de programare vizuală dezvoltat de MIT Media Lab. Scratch a fost creat în 2006 și este folosit în majoritatea școlilor din lume. Este disponibil în mai mult de 70 de limbi.

Scratch 2 este disponibil și în versiunea offline. Poate fi descărcat de la adresa: <https://scratch.mit.edu/download>. Scratch 2 nu este disponibil pentru tablete.

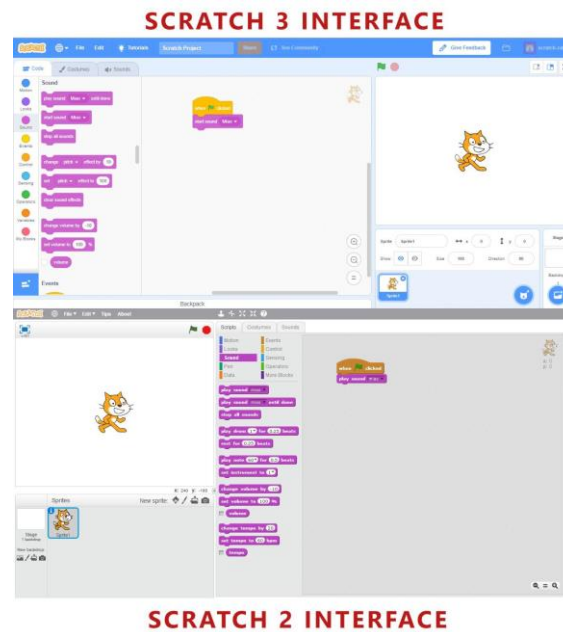
Există o serie de diferențe între versiunile de Scratch 2 și 3. Ambele versiuni pot fi folosite online prin intermediul unui browser web pe PC-uri. Scratch 3 e dezvoltat în HTML5, așa că poate rula și pe tablete Android sau iPad-uri, dar nu are o versiune offline, precum Scratch 2 care e dezvoltat în Flash. Ambele programe pot interacționa cu dispozitive externe, dar numai pe calculatoare, deoarece e necesară instalarea unui mic program (linker). Scratch 2 poate controla, în mod nativ, Lego WeDo 1 și 2 și Picoboard. Scratch 3 poate controla, în mod nativ, Lego WeDo 2, Lego Mindstorm EV3 și Microbit. Scratch 3 are funcții în plus, precum „text to speech” în diferite limbi și translator. E posibilă, de asemenea, construirea de noi extensii, astfel, în viitor, probabil vor fi disponibile și mai multe funcții.

Există și o serie de funcții adiționale comune celor două versiuni cum ar fi unealta de desen, instrumente muzicale și senzor video.



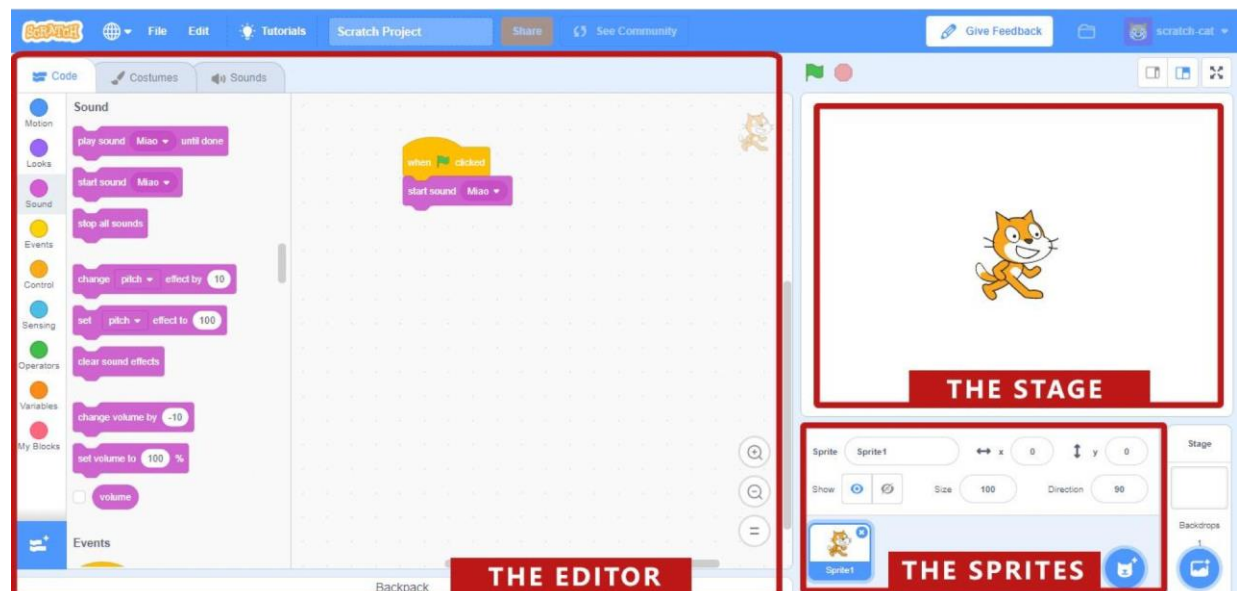
Principalele diferențe dintre Scratch 2 și Scratch 3		
	Scratch 2	Scratch 3
Versiuni Offline (Windows, OSX)		
Versiuni Online	(doar pe calculatoare)	(pe calculatoare și tablete)
Dispozitive Externe		
Lego WeDo 1	x	
Lego WeDo 2	x	x
Picoboard	x	
Lego Mindstorm EV3		x
Microbit		x
Funcții Extra		
Text-to-speech în diferite limbi		x
Unealtă de desen (Drawing Pen)	x	x
Instrumente muzicale	x	x
Senzor video	x	x
Translator de texte		x

Ambele versiuni permit salvarea și partajarea proiectelor Scratch în comunitatea Scratch (Scratch Community). Fișierele sunt compatibile cu ambele versiuni. Din ianuarie 2019, singura versiune online este Scratch 3, iar singura versiune offline este Scratch 2.



Interfața

Noua interfață a mediului Scratch 3 este prezentată în imaginea următoare.



Aceasta este împărțită în trei zone principale:

Scena (en. Stage) este cea mai importantă zonă unde programul va “prinde viață”.

Zona personajelor (en. Sprite) este zona în care se găsesc toate componentele (personajele) care fac parte din scenă.

Zona de editare (en. Editor) cuprinde trei tipuri de editoare:

Editorul de cod: conține lista blocurilor disponibile și toate blocurile folosite pentru a descrie comportamentul personajului (sprite-ului) respectiv. Este importantă înțelegerea faptului că fiecare personaj are propriile sale blocuri, astfel încât la selectarea diferitelor personaje din zona de personaje, zona de editare se va schimba.

Editorul de costume: permite desenarea și modificarea aspectului personajelor. Fiecare personaj are propriile sale costume.

Editorul de sunete: permite înregistrarea și editarea sunetelor ce vor fi folosite în program.

Snap for Arduino

Snap for Arduino este o modificare a software-ului Snap!, creat de Universitatea California, Berkeley. Datorită lui Snap (care este dezvoltat în mod constant) putem programa cu ușurință toate plăcile Arduino.



Conform website-ului oficial, Snap4Arduino are următoarele caracteristici:

- e bazat pe blocuri, dinamic, permite programarea în direct, concomitentă, paralelă
- suportă majoritatea plăcilor Arduino
- Folosește firmware-ul standard
- Diagrame cu pini auto-configurabile și abstractizări hardware de nivel înalt
- Permite interacțiunea cu mai multe plăci în același timp
- Versiuni desktop pentru cele trei sisteme de operare principale
- Versiuni online care se pot conecta la plăci Arduino printr-un plugin Chrome
- Software gratuit sub licență Affero GPLv3

- Transformarea unor simple scripturi în schițe Arduino
- Folosește protocolul HTTP pentru controlul la distanță și transmisia live a scenei Snap!
- Versiune în linie de comandă pentru a fi integrată în GNU/Linux

Cum să instalezi cele două software

Pentru a instala Scratch 3: <https://scratch.mit.edu>

Pentru a instala Snap4Arduino: <http://snap4arduino.rocks/>

Scratch 3

Scratch este conceput în special pentru copii cu vârsta între 8 și 16 ani, dar este un instrument foarte util pentru orice începător în programare.

Scratch este utilizat în mai mult de 150 de țări diferite și disponibil în mai mult de 40 de limbi.

Scratch este folosit ca limbaj introductiv, deoarece crearea de programe interesante este relativ ușoară, iar cunoștințele pot fi aplicate altor limbaje de programare, cum ar fi Python și Java.

Categorie	Notițe		Categorie	Notițe
Mișcare	Deplasare personaje, modificare unghiuri și modificare valori X și Y.		Detectare	Personajele pot interacționa cu elementele înconjurătoare pe care le-a creat utilizatorul
Aspect	Controale pentru aspectul personajelor; atașează discurs sau bulă de gândire, modifică fundalul, mărește sau micșorează, transparență, nuanță		Operatori	Operatori matematici, generator de numere aleatorii, operatori logici care compară pozițiile personajelor
Sunet	Pornește fișiere audio sau efecte. Secvențele programabile sunt acum		Variabile	Utilizarea și atribuirea variabilelor și a listelor

		disponibile în categoria Muzică din extensii.			
	Evenimente	Conține elemente de gestionare a evenimentelor plasate în partea de sus a fiecărui grup de blocuri		Blocurile mele	Proceduri personalizate (blocuri).
	Control	Conține instrucțiuni de control cum ar fi: "dacă-atunci" (en. if-else), „la infinit” (en. "forever"), „repetă” (en. "repeat"), și "stop", etc.			

În plus, Scratch include următoarele extensii:

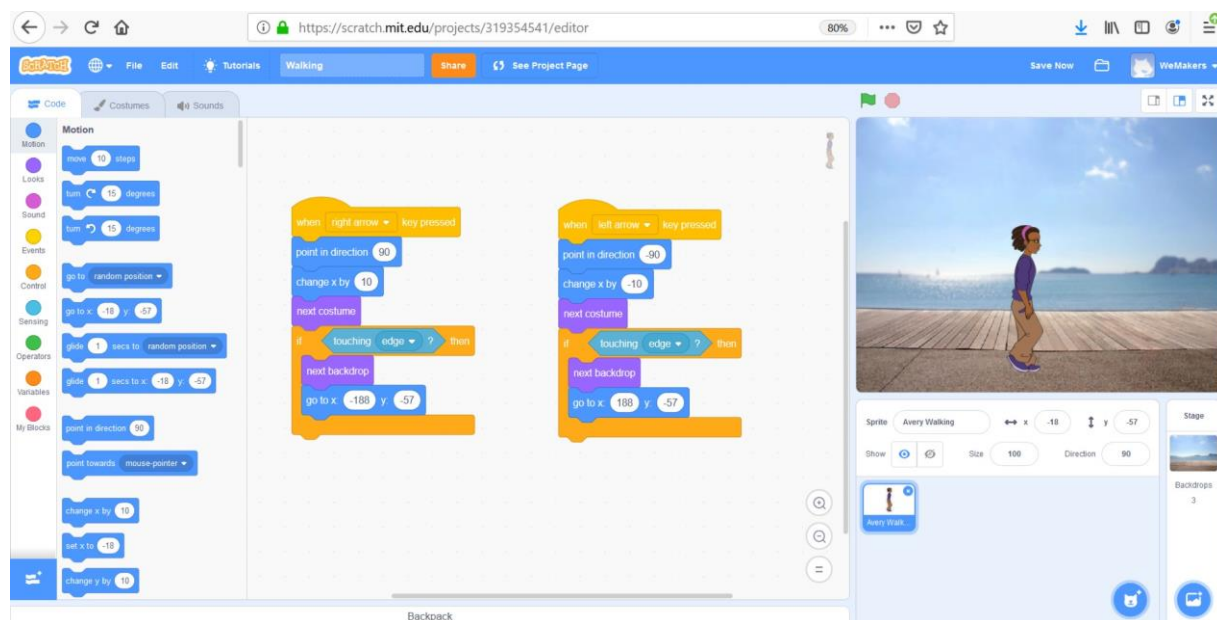
- Muzică (en. Music)
- Stilou (en. Pen)
- Detectare video (en. Video Sensing)
- Text în vorbire (en. Text To Speech)
- Traducere (en. Translate)
- Makey Makey
- micro:bit
- LEGO MIDSTORMS EV3
- LEGO BOOST
- LEGO Education WeDo 2.0
- Go Direct Force & Acceleration

Aplicația 1: un personaj care merge (săgețile stânga/dreapta)

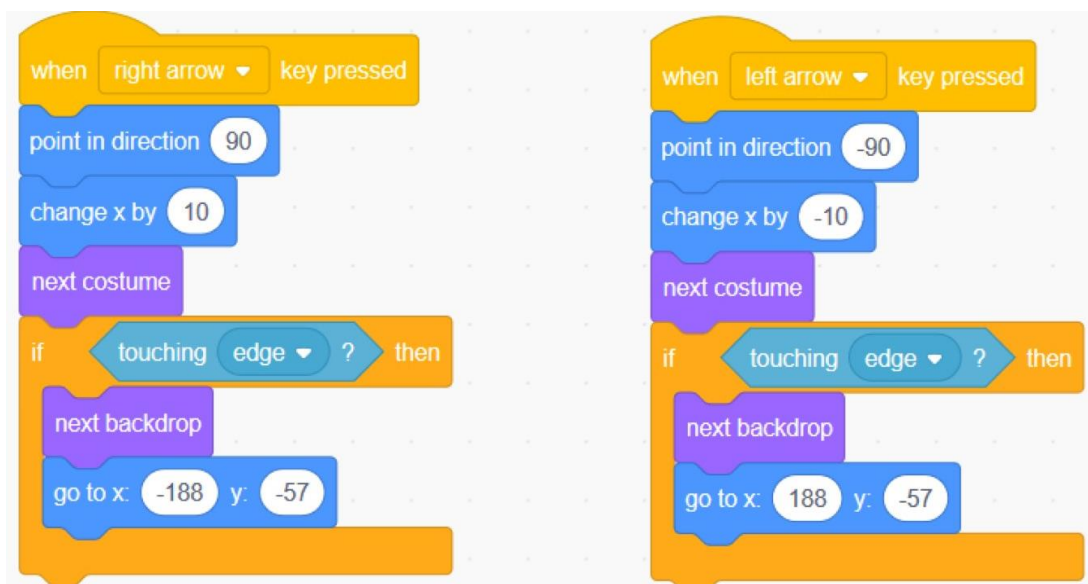
Următoarea aplicație include un personaj (en. sprite) controlat de tastele săgeți stânga și dreapta, care se deplasează în trei sau mai multe decoruri (en. backdrop).

Cum să programăm pas cu pas:

1. Alegeți un personaj (de preferabil unul care are costume potrivite pentru mers/zbor/înot)
2. Alegeți trei sau mai multe decoruri (en. backdrops).



3. Scrieți scripturile prezentate în următoarea imagine
4. Folosiți tastele săgeți stânga și dreapta pentru a testa rezultatul.



Provocări

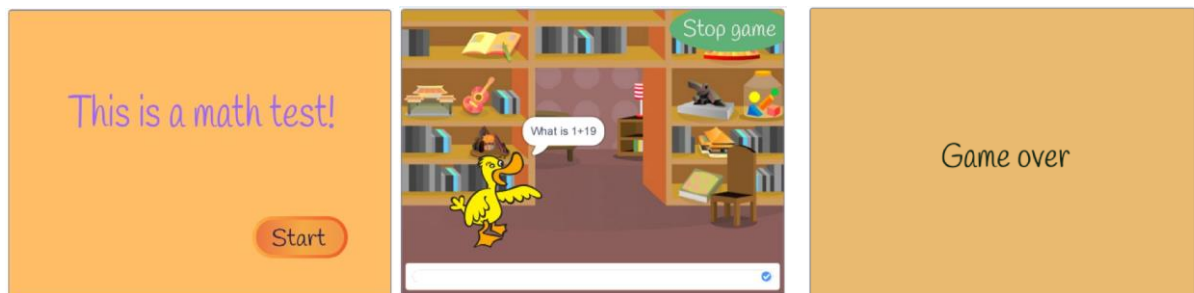
Provocarea 1: Adăugați codul corespunzător pentru a face personajul să sară și să avanseze atunci când se apasă tasta săgeată sus.

Provocarea 2: Creați un joc:

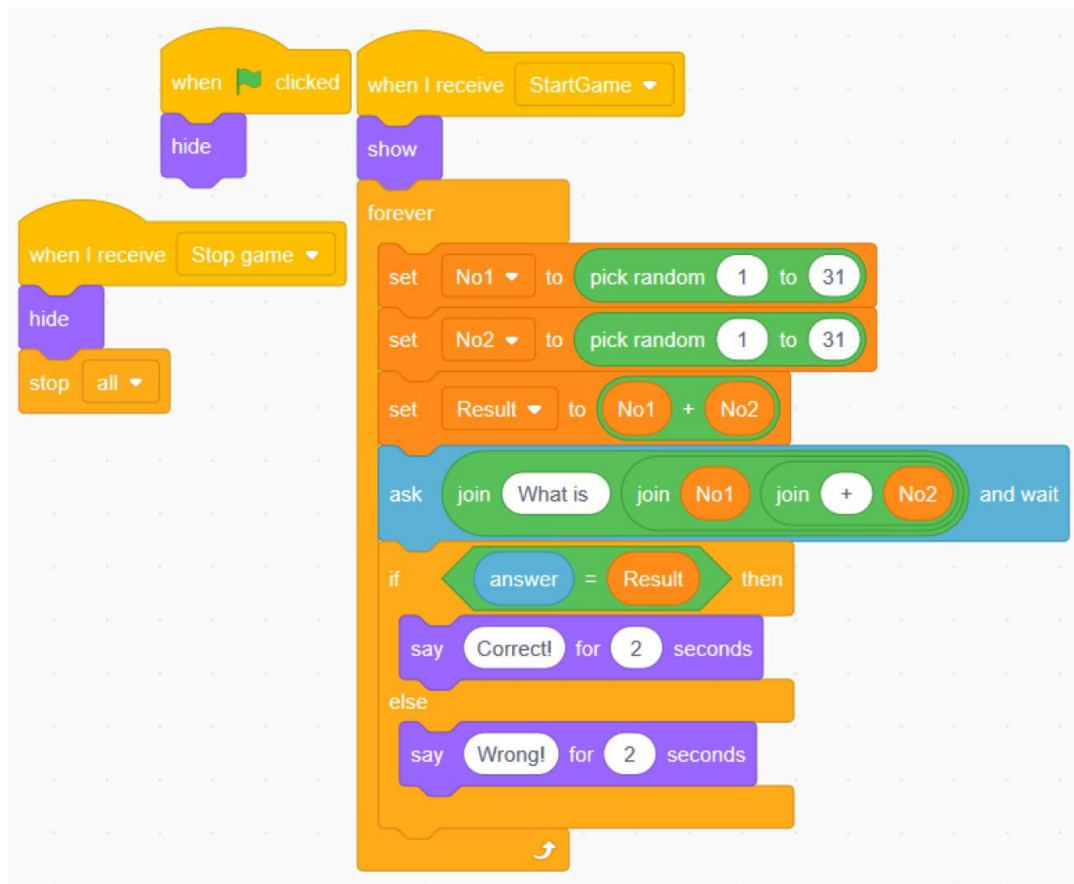
1. un personaj controlabil și o broască ce avansează aleatoriu (se modifică doar valoarea axei x)
2. Personajul nu trebuie să atingă broasca (trebuie să sară peste broască)
3. Are trei vieți
4. Dacă atinge broasca pierde o viață
5. Jocul se termină când pierde ultima viață

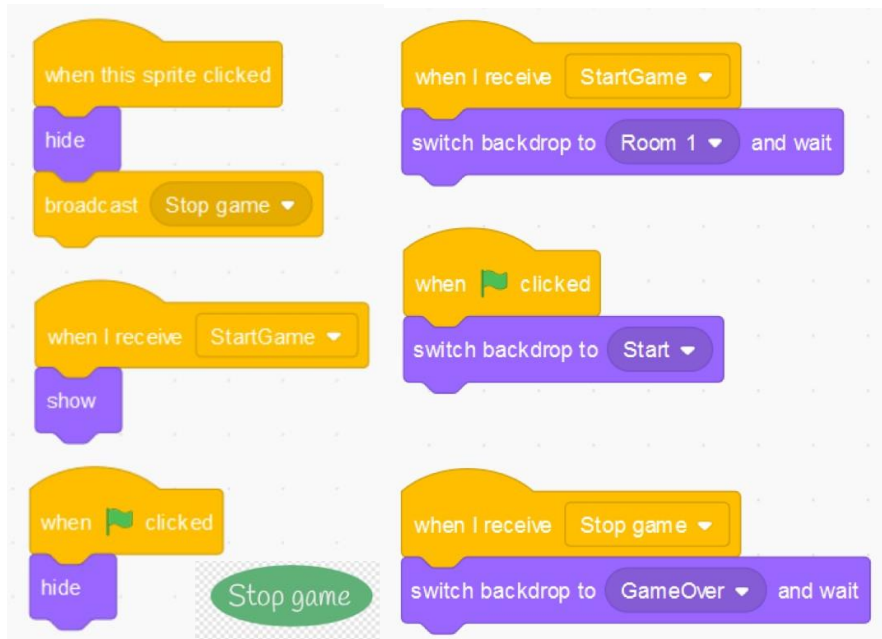
Se poate adăuga și codul care face ca personajul să se schimbe într-o broască pentru 2 secunde atunci când atinge broasca.

Provocarea 3: Jocul generează adunări aleatorii și afișează *Corect* sau *Inc corect* la fiecare răspuns. Indicație: Include trei decoruri și trei personaje (2 butoane și un personaj care adresează întrebările și oferă feedback).



Scripturile pentru fiecare personaj și pentru decoruri:





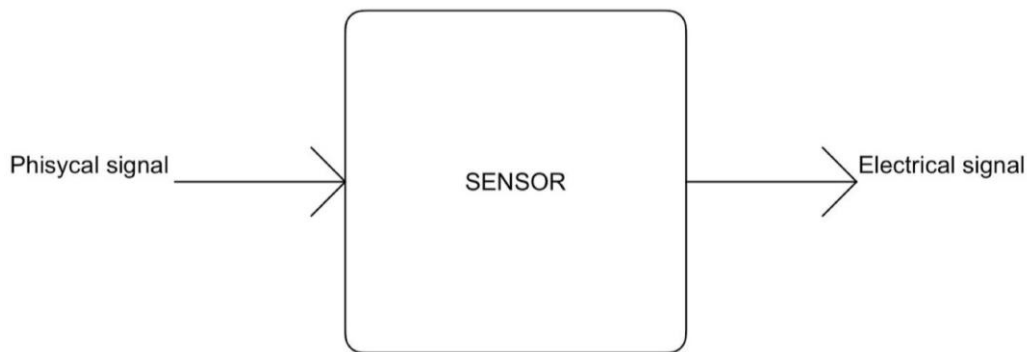
Provocarea 4: Adăugați două variabile care să contorizeze numărul total de întrebări și numărul total de răspunsuri corecte. Afișați cele două valori la sfârșit.



Provocarea 5: Creați propriul joc cu 2-3 întrebări.

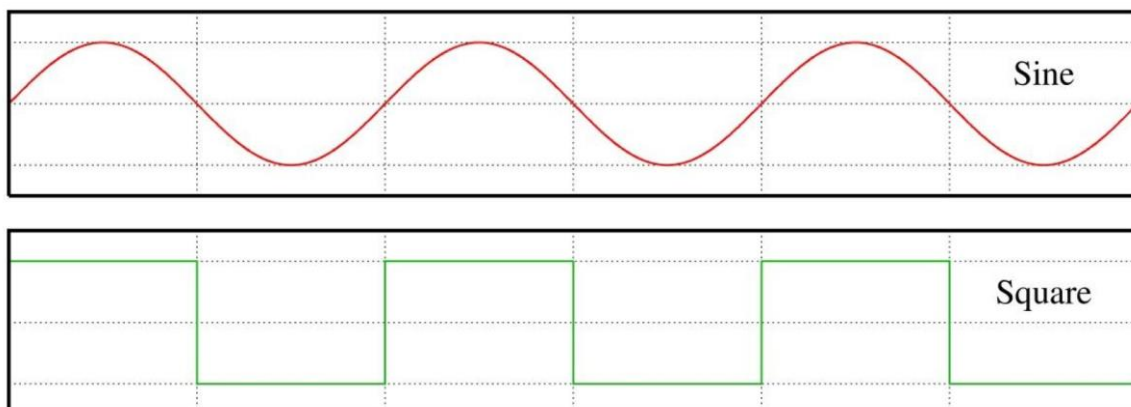
Senzori: o privire de ansamblu

Un senzor e un dispozitiv folosit în electronică pentru a detecta modificarea unui parametru fizic în mediu și trimiterea acestei informații, codificată într-un semnal electric, către alte dispozitive pentru a fi manipulată și analizată.



Semnalul electric poate fi analog sau digital. Analogic înseamnă că semnalul variază continuu, între două valori: minim și maxim. Digital înseamnă că semnalul poate avea doar un număr limitat de valori, în general două: nivelul de jos și nivelul de sus (minim și maxim).

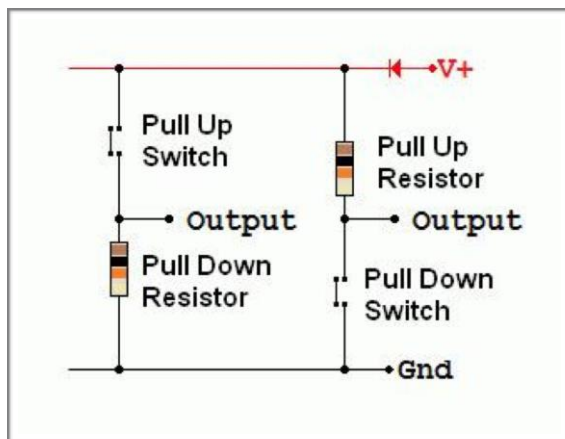
În figura următoare sunt prezentate două exemple de semnale: un semnal sinusoidal și un semnal dreptunghiular. Primul presupune toate valorile de la limita inferioară la limita superioară, deci este un semnal analogic. Cea de-a doua presupune doar două valori, trecând de la o stare joasă la una înaltă și invers, deci este un semnal digital.



Un exemplu de semnal analog îl reprezintă temperatura: poate arăta orice valoare, de la zero absolut la infinit. Exemplul tipic pentru semnale digitale îl reprezintă starea unui buton: poate fi apăsat sau nu. Senzorul analog tipic îl reprezintă

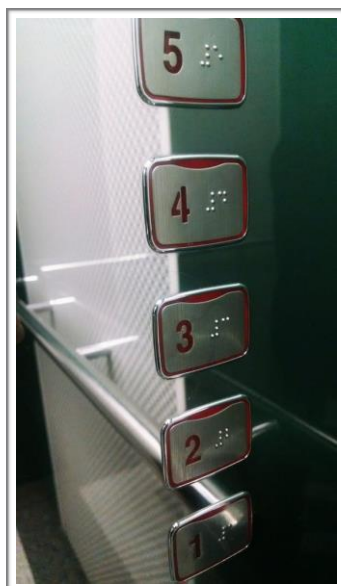
rezistorul variabil (spre exemplu un potențiomtru folosit pentru a modifica volumul unui amplificator stereo) și senzorul digital tipic este butonul, prezentat în secțiunea următoare.

Senzor de atingere



Cel mai simplu senzor pe care îl putem construi este senzorul de atingere. Există diferite tipuri de senzori tactili, dar cel mai simplu îl reprezintă un buton sau întrerupător. Un buton este doar un dispozitiv electromecanic care poate închide sau deschide un circuit electric. Când circuitul este închis, curentul electric poate trece, și când e deschis curentul nu poate trece.

Punând un buton într-un circuit corespunzător cu un rezistor, obținem un circuit care ar putea oferi doar două valori de tensiune, mică sau mare, corespunzător stării butonului. Schema alăturată prezintă conexiunile. Rezistorul și comutatorul pot fi puse în două moduri. În primul mod, cu rezistorul „pull down” se obține la ieșire



valoarea mică când întrerupătorul este deschis și valoarea mare atunci când este închis. În versiunea cu rezistorul "pull up" se obține comportamentul opus. De exemplu, folosim butoane și întrerupătoare pentru a porni și opri luminile sau pentru a selecta etajul potrivit într-un lift.

La următoarea adresă puteți găsi mai multe informații despre butoane și o placă programabilă „open source”: <https://www.arduino.cc/en/Tutorial/Button>

Există și alți senzori de atingere, iar cei mai populari sunt senzorii capacitivi. Tehnologia acestor senzori este folosită în dispozitivele cu ecran tactil (en. “touch screen”).

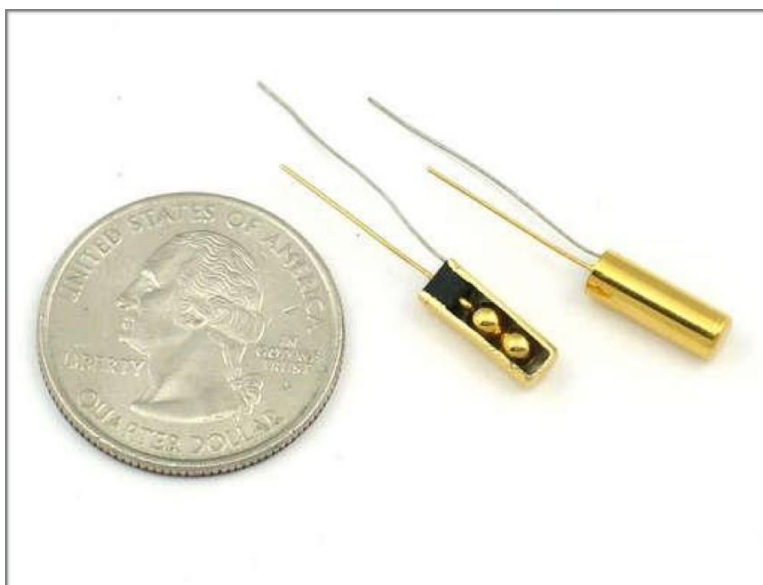
Ei măsoară variația capacității unui condensator determinat de prezența umană. Prin urmare, avem un dispozitiv ce oferă o ieșire cu două niveluri, precum un buton.

Senzor de înclinare

Senzorul de înclinare este folosit pentru a detecta dacă un obiect este înclinat într-o direcție sau alta. Cel mai simplu senzor de înclinare îl constituie comutatorul cu mercur: e compus dintr-un balon ce conține două contacte și o cantitate mică de mercur, liberă să se miște. Când balonul este înclinat, mercurul se deplasează iar când atinge contactele închide circuitul. Mercurul este adesea înlocuit cu o bilă metalică. Dacă se folosesc mai multe contacte e posibilă crearea de senzori de înclinare pe mai multe axe.

Senzorii de înclinare sunt folosiți, spre exemplu, în unele jucării yo-yo care emit lumină atunci când cineva se joacă cu ele.

La următoarea adresă poate fi găsit un exemplu cu un senzor de înclinare cu o singură axă și o placă programabilă: <https://learn.adafruit.com/tilt-sensor/using-a-tilt-sensor>

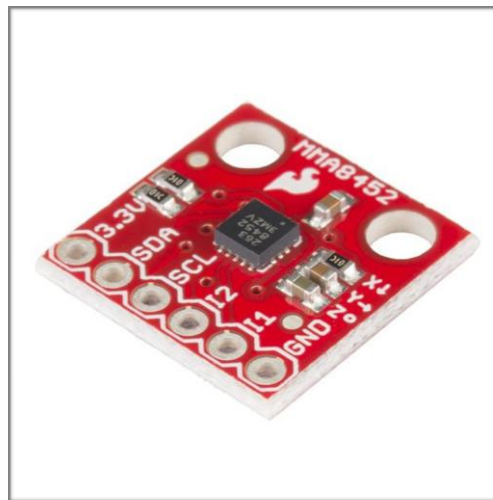


Accelerometru

Accelerometrul este un dispozitiv folosit pentru a măsura accelerația. Acesta funcționează datorită inerției fizice a maselor. Conceptual, un accelerometru este compus dintr-o masă conectată la un arc amortizat. Când accelerometrul este supus unei accelerații, masa se deplasează în raport cu containerul, proporțional cu modulul accelerației. Măsurarea deplasării indică accelerația. Accelerația e apoi codificată într-un semnal electric. Accelerometrele sunt folosite, spre exemplu în

sistemele de navigație pentru a obține poziția relativă față de un punct de referință zero, în telefoanele inteligente, pentru a roti ecranul, pentru a juca anumite jocuri, ca pedometrul și altele, sau în unele calculatoare cu hard disc magnetic ca detector de cădere pentru a preveni problemele asociate cu poziționarea capului de citire într-o poziție sigură.

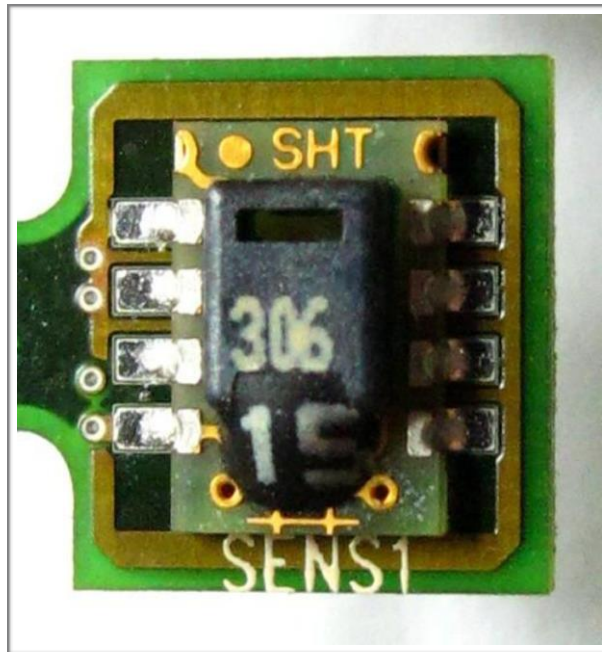
La adresa următoare puteți găsi un exemplu de utilizare a unui accelerometru analog cu 3 axe, care furnizează o tensiune de 3 volți proporțională cu accelerațiile, cu o placă programabilă: <https://www.arduino.cc/en/Tutorial/ADXL3xx>



Senzor de umiditate

Senzorii de umiditate măsoară umiditatea din mediu (ex. din aer sau din sol). În general ei funcționează prin măsurarea variației unui condensator sau a unei rezistențe datorată variației cantității de apă din mediu. Senzorii de umiditate sunt folosiți spre exemplu în meteorologie, de asemenea în stațiile meteorologice portabile și în sistemele automate de irigare care pot porni apa când solul este uscat.

La adresa următoare poate fi găsit un exemplu cu o placă programabilă open source și un senzor de umiditate comun (care poate măsura și temperatura): https://www.tutorialspoint.com/arduino/arduino_humidity_sensor.htm



Senzor de temperatură

Un senzor de temperatură convertește temperatura într-un semnal electric. Acest lucru e posibil datorită mai multor fenomene fizice, în funcție de tipul de senzor. Spre exemplu termistorul prezintă o variație a rezistenței sale în funcție de variația temperaturii și termocuplul prezintă o tensiune proporțională cu diferența de temperatură.

Senzorii de temperatură sunt utilizați în meteorologie, termometre medicale, procesoarele calculatoarelor, etc.

La următoarea adresă puteți găsi un exemplu de citire a temperaturii cu ajutorul unei plăci programabile:

<https://learn.adafruit.com/tmp36-temperature-sensor/using-a-temp-sensor>

Ieșiri: o privire de ansamblu

O ieșire este un element care poate acționa în lumea reală folosind diferite moduri de interacțiune. Cele mai importante ieșiri produc mișcare, lumini sau sunete pentru a interacționa cu utilizatorii și cu mediul.

LED

Dacă doriți să introduceți LED-urile în clasa voastră, puteți folosi mai multe metode diferite, în funcție de tipul școlii. În acest manual vă vom prezenta generic toate componentele electronice de care aveți nevoie, astfel încât profesorii care predau diferite discipline să le poată folosi fără probleme.

Înainte de a defini o componentă, trebuie să legăm creația componentei de viața inventatorului acesteia. Inventatorul LED-ului este Nicholas Holonyak Jr, un american cu origini rusești, care a dezvoltat primul LED în 1962. El a reușit să emită doar lumină roșie. În zilele noastre se găsesc de cumpărat LED-uri de culori foarte diferite. De exemplu, datorită lui Asaki, Amano și Nakamura s-au creat LED-uri albastre. Această cercetare a fost atât de importantă încât au câștigat un premiu Nobel.



Un LED este o ieșire aparent foarte simplă, care permite să emită lumină de culori diferite într-un mod fiabil, cu o durată de viață mare la costuri reduse. Astăzi LED-urile sunt utilizate pentru multe aplicații, nu numai în scopuri comerciale, ci și în scopuri casnice.

Principiul de funcționare se bazează pe electroni care ocupă unele goluri dintr-un material semiconductor și eliberează energie, sau fotoni. În funcție de material se

emit frecvențe și culori diferite. LED-ul are nevoie de curent constant pentru a fi alimentat, acesta este motivul pentru care este folosit întotdeauna cu o rezistență care permite controlul fluxului electric care ajunge la diodă.

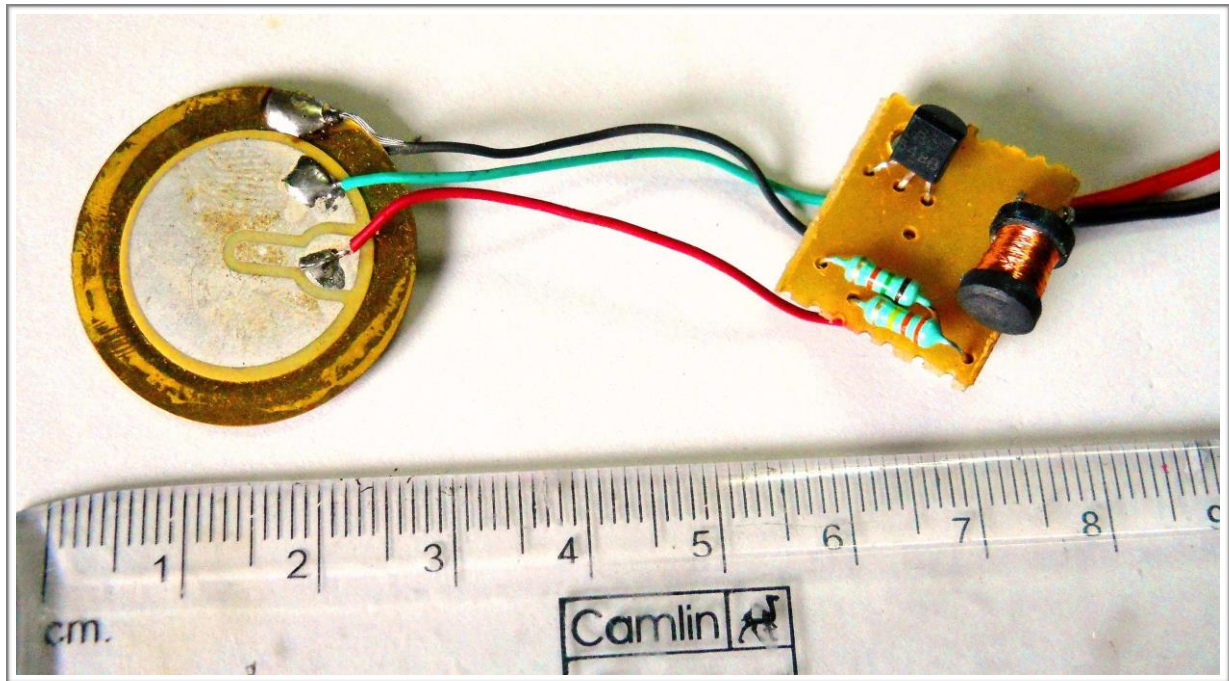
Mai multe exerciții:

În continuare e prezentată o listă de exerciții suplimentare (ce nu sunt detaliate în acest manual) ce vor ajuta elevii să aplice și să-și îmbunătățească cunoștințele:

- Modificarea timpului de clipire a LED-ului, de la 1 la 3 secunde.
- Modificarea pinul digital (nu 0 și 1!) la care este conectat LED-ul și schimbarea programului în mod corespunzător.
- Modificarea timpului de clipire după cum urmează: LED-ul pornit pentru 1.5 secunde – LED-ul oprit pentru 2/3 secunde.

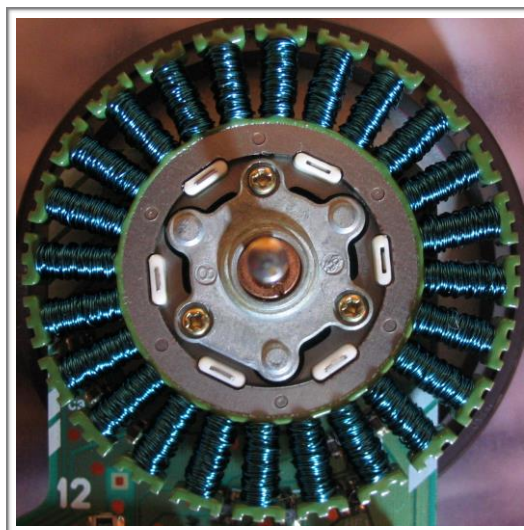
Buzzer

Un prim lucru pe care îl puteți face este să folosiți Arduino pentru a emite sunete cu diferite tonuri și durate diferite. Astfel puteți vorbi cu elevii despre muzică și cântecele preferate ale acestora. În plus puteți viziona împreună cu elevii, din filmul lui S. Spielberg "Întâlnire de gradul trei" (en. "Close Encounters of the Third Kind"), partea în care oamenii comunică cu extraterestrii cu ajutorul a 5 note muzicale. Desigur, le puteți cere elevilor să creeze propria muzică. În acest fel, puteți implica elevii să utilizeze programarea într-un mod creativ. După ce ați introdus modul de emiteră a notelor cu buzzerul, puteți crea programe pentru a cânta melodii celebre.



Motor

Motoarele permit deplasarea unui robot pentru prima dată. Prima mișcare va fi simplă și amuzantă: robotul va înainta fără să țină cont de obstacole și timpul de operare. În lumea Arduino, H-bridge este un tranzistor ce ne permite să controlăm simultan două motoare de curent continuu. Fără H-bridge nu ar fi posibilă deplasarea robotului. H-bridge este de fapt o baterie care gestionează alimentarea cu energie a roboților. Motoarele incluse în cele mai populare kit-uri Arduino sunt motoare de curent continuu fără niciun senzor. De aceea nu este posibilă gestionarea rotației motoarelor, ci doar curentul livrat acestora.



“Fiecare realizator de jocuri video știe ceva ce realizatorii de curriculum pare că nu reușesc să-l înțeleagă. Nu vei vedea niciodată un joc video promovat ca fiind ușor. Copii cărora nu le place școala nu vor spune că nu le place pentru că e prea grea, ci pentru că e plictisitoare.”

— Seymour Papert

Provocări

În acest capitol sunt prezentate câteva activități (provocări) ghidate cu diferite niveluri de dificultate și care implică diferite componente hardware și software.

Fiecare provocare reprezintă o sarcină ce poate fi rezolvată în clasă, folosind câteva metodologii educaționale:

- metodologia 4P: Joacă, Pasiune, Lucru în grup, Proiect (en. 4P: Play, Passion, Peer, Project)
- Învățarea bazată pe proiect (en. PBL: Project Based Learning)
- Clasa răsturnată (en. Flipped Classroom)

În continuare sunt prezentate câteva definiții scurte ale fiecărei metodologii din cele ce vor fi aplicate în cadrul provocărilor noastre educaționale:

4P: Project, Play, Passion and Peer

Mitchel Resnick a introdus în cartea *Kindergarden: LifeLong learning* conceptul de *elev creativ*.

Un *elev creativ* are nevoie să rezolve probleme concentrându-se pe proces.

Proiecte:

Un proiect este un proces care poate rezolva o serie de aspecte ale problemei pe care dorim să o rezolvăm.

Nu avem nevoie de un proiect concret. De asemenea, putem folosi teoria pentru a înțelege lumea și de a găsi o propunere de îmbunătățire a uneia sau a mai multor probleme.

Lucru în grup:

Colaborarea este fundamentală în procesul de învățare. Procesul de predare colegilor, împărtășirea informației și ascultarea colegilor ne permite îmbunătățirea cunoștințelor și a conștientizării cu privire la prieteni și audiență. Datorită acestei colaborări ne dezvoltăm empatia.

Pasiune:

Înseamnă implicare. Putem lucra cu emoțiile pozitive ale elevilor noștri când folosim

tehnologia.

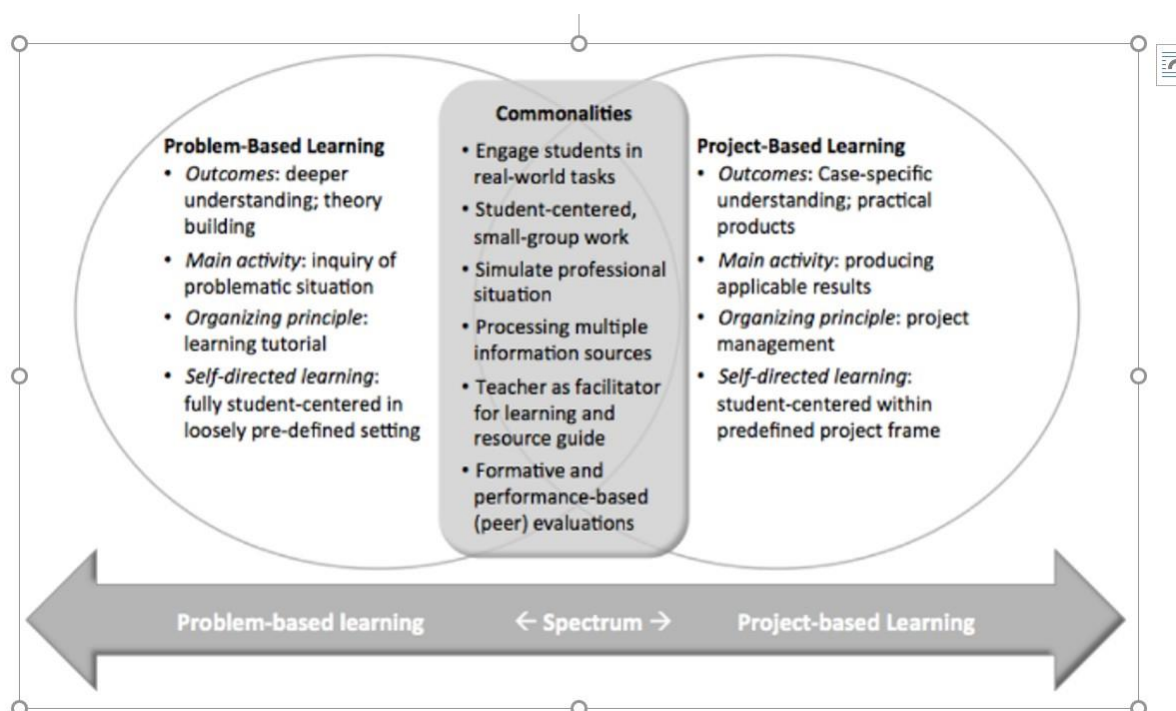
Joacă:

Avem nevoie de distracție în procesul de învățare. Un simplu mediu, fețele vesele permit oamenilor să învețe mai bine și mai repede.

Învățarea bazată pe proiect

Pentru a încuraja elevii creativi, *Învățarea bazată pe proiect* este o metodologie bazată pe studiul unui proiect care implică toate disciplinele. Acest lucru înseamnă planificarea de scenarii din lumea reală pentru elevi în care ideile să fie conectate la rezolvarea de probleme reale. Mulțumită metodologiei de învățare bazate pe proiect, elevii sunt mai implicați și își pot îmbunătăți colaborarea, abilitățile fine, favorizează creativitatea și învățarea e mai distractivă!

Pe wikipedia putem găsi o imagine care face mai ușoară înțelegerea acestei metodologii:



Clasa răsturnată

Această metodologie vine de la metodele de învățare combinate și presupune o schimbare a clasei, unde elevii trebuie să studieze diferite teme și să-i învețe apoi pe colegi. În acest proces, profesorul își poate schimba rolul și să devină un fel de

antrenor, organizator al cursanților care îi ajută pe aceștia să descopere lucruri noi, prin lucru în grup și prezentări.

Provocarea 1: Bagheta magică

Kituri robotice: Lego Wedo 2 sau Arduino sau Microbit (în continuare se prezintă varianta cu Lego Wedo 2)

Software: Scratch 3.0 și Lego Digital Designer

Vârsta: de la 10 la 15 ani

Introducere

Trebuie să asamblăm o baghetă cu piese Lego WeDo 2. Este foarte simplă și nu necesită manuale. Din acest motiv, din punct de vedere educațional, lecția poate fi organizată pornind de la proiectul baghetei atât pe foi, cât și folosind software-ul gratuit Lego Digital Designer.

În această fază elevii vor trebui să:

1. proiecteze o baghetă pe hârtie sau calculator (cu LDD)
2. realizeze cu piese Lego WeDo bagheta proiectată, plasând senzorul de înclinare în mânerul baghetei.

Scopul exercițiului este de a capta atenția și implicarea elevilor folosind tehnici narative (povești cu magicieni). Datorită acestei activități se pot introduce utilizarea senzorului de înclinare și înțelegerea utilizării instrucțiunii “if” în Scratch.

Povestire

Exercițiul permite integrarea lui într-un scenariu cu magi, în care elevii sunt magicieni protagoniști. Ei pot pregăti pălării de magicieni pe care să le poarte pe parcursul desfășurării lecției. Profesorul poate prezenta povești cu magicieni folosind vrăjile de programare următoare ca o fază din scenariul lecției.

Primul Program

Primul program e folosit doar pentru a mânui bagheta și pentru a schimba costumul unui personaj introdus în Scratch atunci când bagheta este mișcată. Spre exemplu, schimbarea poziției baghetei de la poziția verticală la cea orizontală ar putea să conducă la modificarea formei personajului din Scratch.

Nu sunt necesare cunoștințe anterioare de programare pentru a rula acest program. Prin acest exercițiu simplu, elevii pot descoperi două blocuri importante din programare:

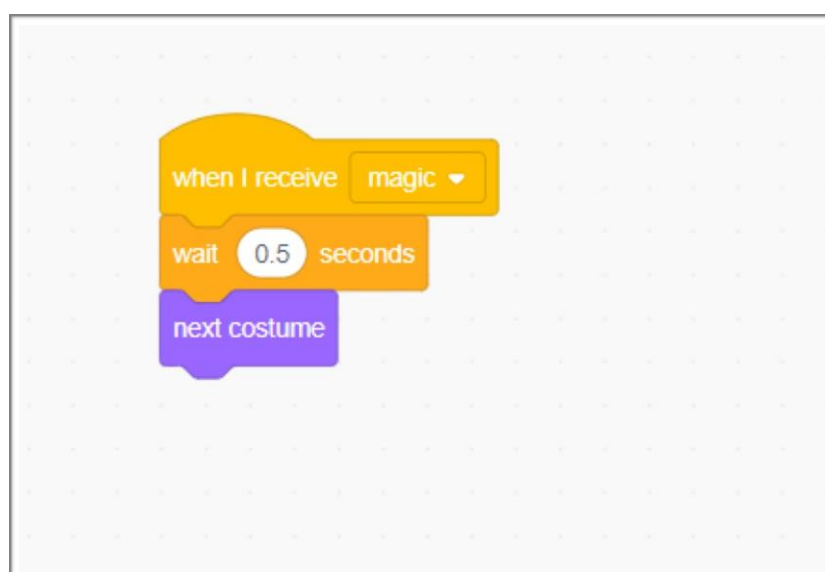


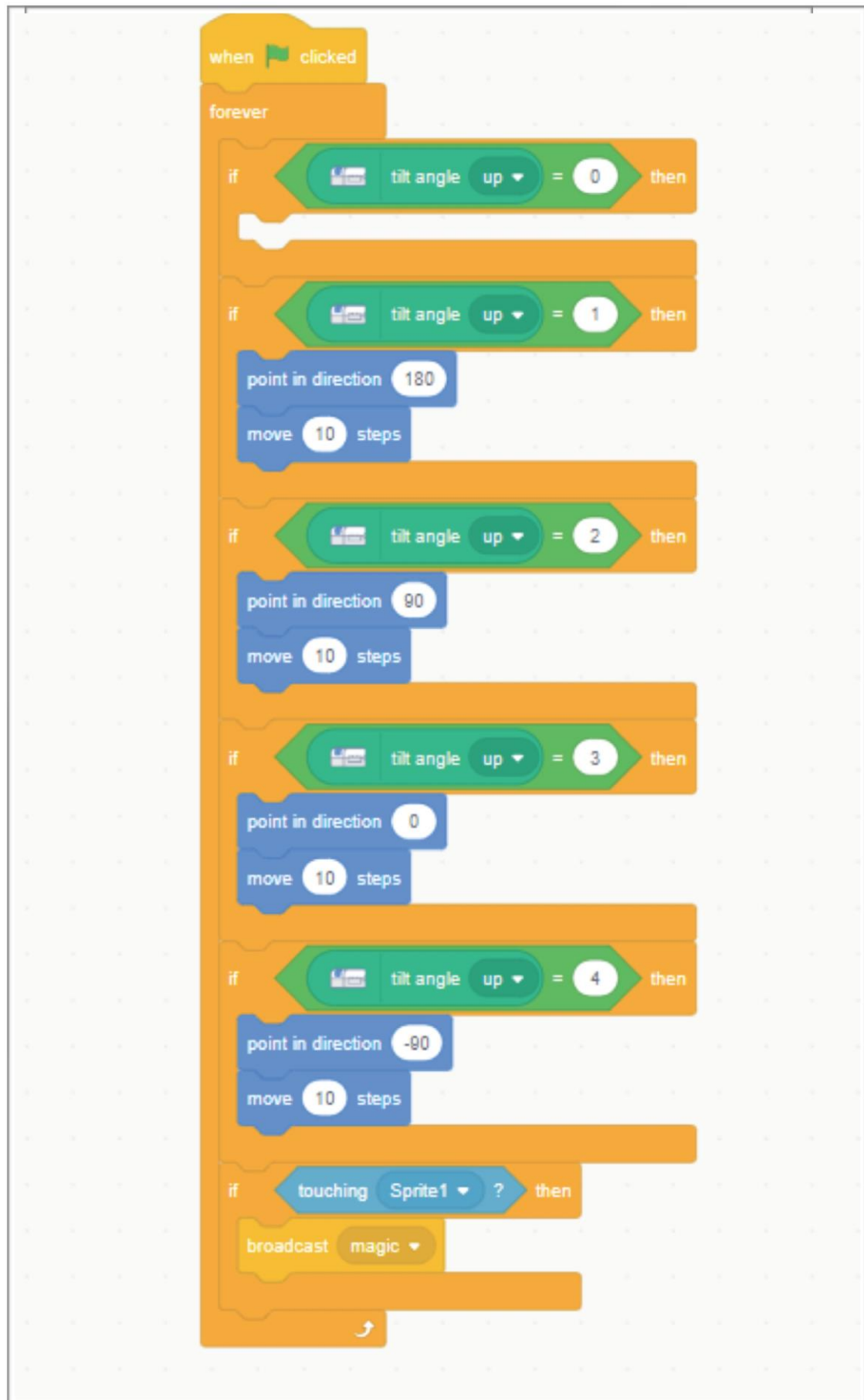
Acest bloc *forever* (ro. *la infinit*) permite crearea unei structuri repetitive ce se repetă la infinit. În acest caz acest bloc ne permite să controlăm permanent poziția baghetei.

Al doilea bloc important este blocul *if-then* (ro. *dacă-atunci*) care permite compararea poziției senzorului, și în funcție de unghi putem programa schimbarea imaginii de pe ecran.



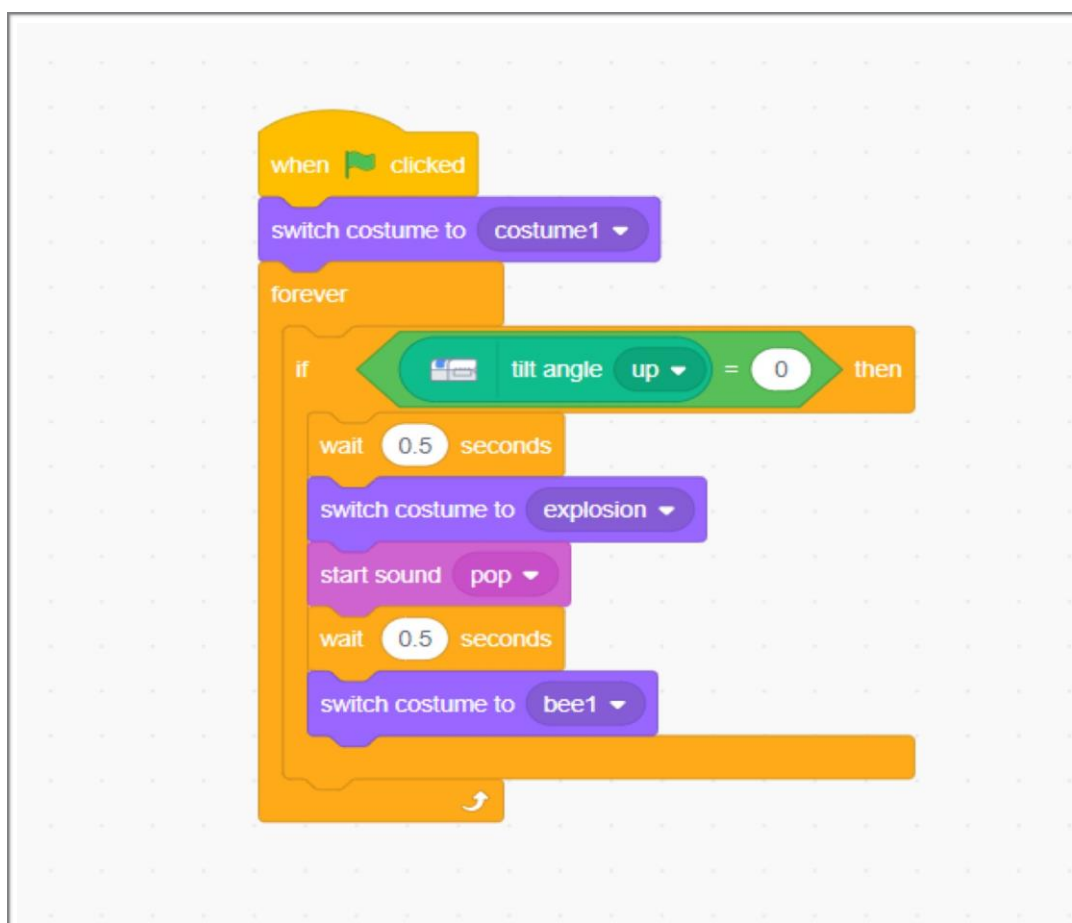
Datorită acestui program simplu, elevii vor putea înțelege potențialul comenzii “if” și să construiască o diagramă simplă care să includă una sau mai multe opțiuni bazate pe starea senzorului de înclinare (en. TILT) a setului Lego WeDo.





Program 2

Al doilea program nu vizează extinderea orizontului cunoașterii, ci consolidarea noilor concepte învățate. Al doilea exercițiu prevede ca, la momentul executării magiei, să se audă un sunet și să se realizeze un efect grafic care anunță magia. Din punct de vedere didactic nu apar concepte noi, astfel încât profesorii pot folosi acest exercițiu pentru a-și da seama dacă elevii au înțeles potențialul aplicației.



În plus, această parte a programului permite o personalizare realizată de către elev (nu doar selectarea unui sunet dar și realizarea efectului magic). Această personalizare permite o mai mare implicare a elevilor și o învățare mai bună și imediată.

Program 3

În primele două exerciții se pot realiza numeroase variații pe subiect, de dificultăți asemănătoare. Pentru a crește nivelul de programare profesorul poate cere elevilor: “creați o baghetă magică care devine un controler a unei stele într-un video. Tot ce este atins de baghetă este transformat”.

Această cerință implică crearea de noi algoritmi:

1. deplasarea unui personaj în funcție de poziția senzorului de înclinare (4 stări identificate);
2. crearea de legături între fluxurile de programare (a două personaje diferite) folosind comanda "broadcast" (ro. *difuzează*).

Provocarea 2: Peștele și senzorul de înclinare

Scopul este de a aplica toate cunoștințele acumulate până acum într-un program complex, ce implică interacțiunea dintre mai multe personaje. Variabila utilizată (deja prezentă în Scratch) are numele “dimensiune”. Această parte e dedicată, în special elevilor din clasa a cincea. Structura prezentului material de prezentare a acestei activități e puțin diferită față de cea de la activitățile anterioare, și anume o descriere pas cu pas.

Scop

Obiectivul este de a crea un joc în care un pește prădător (doar digital sau chiar “real”), comandat de senzorul de înclinare, poate mânca doar peștii mici. Cu fiecare pește mâncat, peștele protagonist, crește în dimensiune. Inamicul este un pește mult mai mare, un super-prădător, pe care protagonistul trebuie să-l evite până devine suficient de mare încât să nu mai poată fi mâncat de super-prădător. Doar atunci se va ajunge la sfârșitul jocului.

Povestire

Pentru a construi acest joc video, profesorul trebuie să introducă unele elemente legate de construirea intrigii unei povești. Trebuie să introducem figura protagonistului, scopul jocului (mâncarea altor pești) și figura adversarului, inamicul, care împiedică atingerea obiectivului. Acesta poate fi un prim pas către introducerea propriu-zisă a unei povești în care toți actorii sunt clar definiți: protagonist, ajutor al protagonistului, adversar, ajutor al adversarului, obiect magic, obiectivul care trebuie atins.

Jocul video

Pentru a putea realiza jocul, în continuare sunt prezentați pașii pentru programarea fiecărui element și o imagine a jocului final.

Script

Fundal (Scenă)

Se poate importa și edita un fundal sau se poate crea unul nou. În cadrul acestui exercițiu nu este necesară asocierea unei acțiuni cu fundalul, dar ar putea fi posibilă alternarea de diferite fundaluri, la momente potrivite, planificând modul de apariție.

Personajul protagonist - Deplasarea

Pentru a controla personajul principal doar se copiază în pagina de scripturi a personajului, scriptul creat la programul 3.

Personajul protagonist – Inițializarea variabilei "size"

Acest concept stă la baza oricărui cod ce prevede inițializarea variabilelor programului. În acest caz, variabila pe care o vom folosi este deja definită de Scratch (nu trebuie să o creăm noi înșine) și se numește size (ro. *dimensiune*). Inițializarea se face folosind comanda: "Set size at%" din directorul Looks (ro. *Aspect*), din stânga sus. Setăm variabila "size" la 30%.

Personajul pește mic – deplasare și interacțiune

În jocul video vor fi mulți pești mici, toți având același script. Se creează un singur personaj și se multiplică de câte ori se dorește. Toate personajele din joc trebuie activate prin același buton, în acest caz "Space" (acestea sunt acțiuni activate în același timp). Atunci când se programează un joc video trebuie avut în vedere că starea "show" (ro. "*arată*") sau "hide" (ro. "*ascunde*") se stabilește la inițializare, în special în cazul în care în timpul jocului se dorește utilizarea comenzii „hide”.

Apoi continuăm cu programarea deplasării și interacțiunii. Deplasarea peștilor are loc pe o linie dar trebuie avut în vedere că e necesară comanda "if on edge, bounce" (ro. "*dacă atinge marginea, ricoșează*"), pentru a garanta continuitatea deplasării. Dacă atunci când ricoșează, personajul se întoarce cu partea superioară în jos, e suficientă selectarea opțiunii set „rotation style left-right”.

Personajul principal

Pentru a programa corect trebuie avut în vedere că la momentul în care personajul este atins de un pește, trebuie verificat dacă peștele este mai mare sau mai mic. Dacă protagonistul întâlnește un pește mai mic (<30 în setările noastre...), verificare ce se poate face cu comanda „When I receive... size xx”, e suficientă programarea creșterii dimensiunii protagonistului cu un procent la alegere (spre exemplu cu 10 procente).

Personajul – Pește mare

Ca și în cazul peștelui mic, acesta trebuie să fie vizibil de la început. Spre deosebire de peștii mici, acesta va apărea într-o poziție aleatorie (x, y), gestionată de o

combinație de numere aleatorii ce vor fi inserate cu scriptul corespunzător.

Aleatorie va fi și direcția de deplasare a peștelui. Dimensiunea inițial a peștelui mare va fi setată la 70%. Pentru a garanta deplasarea variată prin spațiul disponibil, se poate seta ca peștele să sară înapoi la fiecare margine cu o viteză aleatorie (numărul de pași e generat aleatoriu în intervalul 1-15). Ca și în cazul peștelui mic, dacă peștele mare atinge personajul principal, acesta trebuie să trimită un mesaj cu dimensiunea sa (în acest caz 70%).

Personajul principal – primirea mesajului de la peștele mare

Atunci când protagonistul primește mesajul de la peștele mare, trebuie evaluat dacă acesta va fi mâncat sau dacă îi va face față. Ultima variantă se va realiza atunci când dimensiunea personajului principal va fi cel puțin de 60%. Dacă se verifică condiția, atunci jocul se termină cu mărirea la maxim a peștelui protagonist. În cazul în care personajul principal e mai mic decât peștele super-prădător jocul se termină cu un text reprezentat de un personaj activat la recepționarea mesajului “Sfârșitul jocului” (en. game over).

Personajul Sfârșitul jocului - Script

Sfârșitul jocului e un personaj creat de utilizator. Textul Sfârșitul jocului e activat atunci când primește mesajul Sfârșitul jocului. Acest personaj este invizibil la începutul programului.

Personajul Peștele cel mare - Victorie

Protagonistul, în cazul victoriei, trimite un mesaj “Am câștigat” către toate personajele. Atunci când mesajul ajunge la personajul Peștele cel mare, acesta trebuie să apară. Dacă se dorește, se poate adăuga un personaj “Ai câștigat!”.

Recapitulare

Exercițiul poate fi găsit aici: [LINK](#)

Realizarea acestei aplicații e mai complexă. Astfel pentru a nu încărca excesiv explicațiile nu am introdus notații didactice în text, ca în cazul anterior. Profesorul care va aplica la clasă acest exercițiu va trebui să clarifice câteva concepte cheie.

Variabila "size"

Scratch permite crearea de noi variabile, dar există și câteva deja definite. "Size"

(ro. *dimensiune*) e un exemplu clasic de variabilă locală, ce poate fi aplicată doar asupra unui personaj pentru care este creată, spre deosebire de alte variabile ce pot fi utilizate cu mai multe personaje.

Utilizarea acestei variabile nu permite spre exemplu invocarea ei în scripturile altor personaje.

Datorită variabilei "size" se poate explica conceptul de variabilă. În acest joc modificarea variabilei size descrește sau nu succesul jocului. Adesea, atunci când se predau variabilele la școală, se folosesc exemple create ad hoc și nu foarte utile sau exemple prea complexe. Folosirea în cadrul acestui joc video, furnizează o abordare facilă a variabilelor.

Numere aleatorii

Pentru a crea peștele ce se deplasează cu viteze aleatorii, e nevoie de generatorul de numere aleatorii din Scratch. Acesta permite profesorilor să-i învețe pe elevi cum sunt folosite aceste generatoare în domeniul calculatoarelor.

Trimitere - Primire

În acest program, comunicarea dintre personaje e esențială. Aceasta permite profesorilor introducerea conceptelor de programare paralelă și fluxuri de programare.

Provocarea 3: Oracol

Software: Scratch 3

Această provocare este legată de domeniul asistenților vocali. Aceștia sunt un produs important pe piață și majoritatea companiilor mari produc proprii asistenți vocali.

La începutul lecției profesorii pot capta atenția elevilor cu un film SF, din 2001, Odissea Spațială. După prezentarea unor asistenți vocali importanți în filmul lui Kubric, este prezentat asistentul HAL2001 care poate interacționa cu astronautii și poate controla nava spațială. Profesorul poate aminti și de filmul Star Trek, în care echipajul poate comunica cu "nava" cerând ajutor, sugestii și asistență.

Ulterior se poate crea o conexiune cu lumea reală prin prezentarea unor produse reale, de pe piață, cum ar fi:

Alexa (Amazon)



Google Home (Google)



Siri



Toți acești asistenți vocali pot răspunde la întrebările utilizatorilor clienților, căutând informații pe internet și utilizând aplicații cum ar fi platforme online de muzică, servicii cloud, piață online.

Din punct de vedere etic, se poate discuta cu elevii studiul de caz legat de utilizarea

vocii unui cântăreț cunoscut (așa cum se întâmplă în *Black mirror*). Utilizarea vocii unui cântăreț faimos creează câteva cazuri de utilizare necorespunzătoare a acestui asistent vocal care substituie prietenii reali ai adolescenților.

Din punct de vedere etic e importantă discutarea acestui caz pentru a introduce utilizarea etică a tehnologiei.

Introducerea asistentului vocal poate fi folosită pentru a discuta și despre concepte de inteligență artificială.

Asistentul vocal și Internetul obiectelor

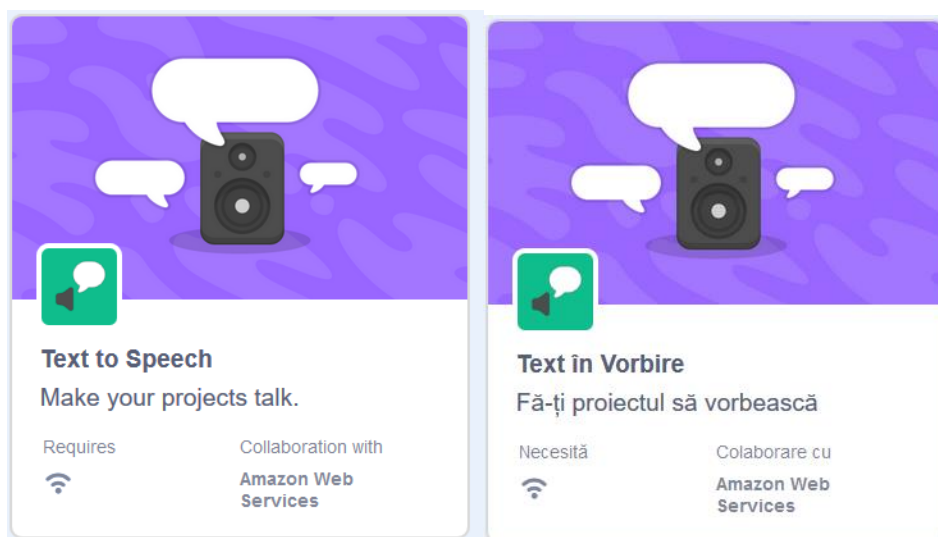
Se prezintă asistenții vocali separat dar în viitorul apropiat toți acești asistenți ar putea reprezenta interfața principală a obiectelor conectate în Internet, în viața noastră de zi cu zi.

După această introducere se va descoperi cum se poate programa un asistent vocal cu Scratch 3.

În Scratch nu e posibilă recunoașterea vocală, dar putem crea un program în care interacțiunea se va face vocal din partea calculatorului și textual din partea utilizatorului.

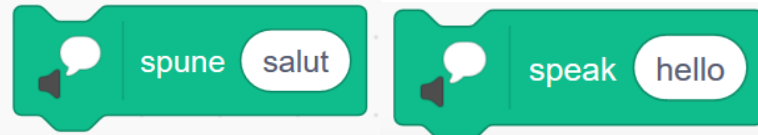
Pentru a programa un asistent vocal, un fel de oracol, se poate folosi comanda Text to speech (ro. „Text în Vorbire”).

Pentru a folosi această funcție, trebuie să apăsăm pe *Add extension* (ro. „Adaugă o extensie”) din colțul din dreapta jos a ferestrei și să selectăm extensia care are imaginea următoare:

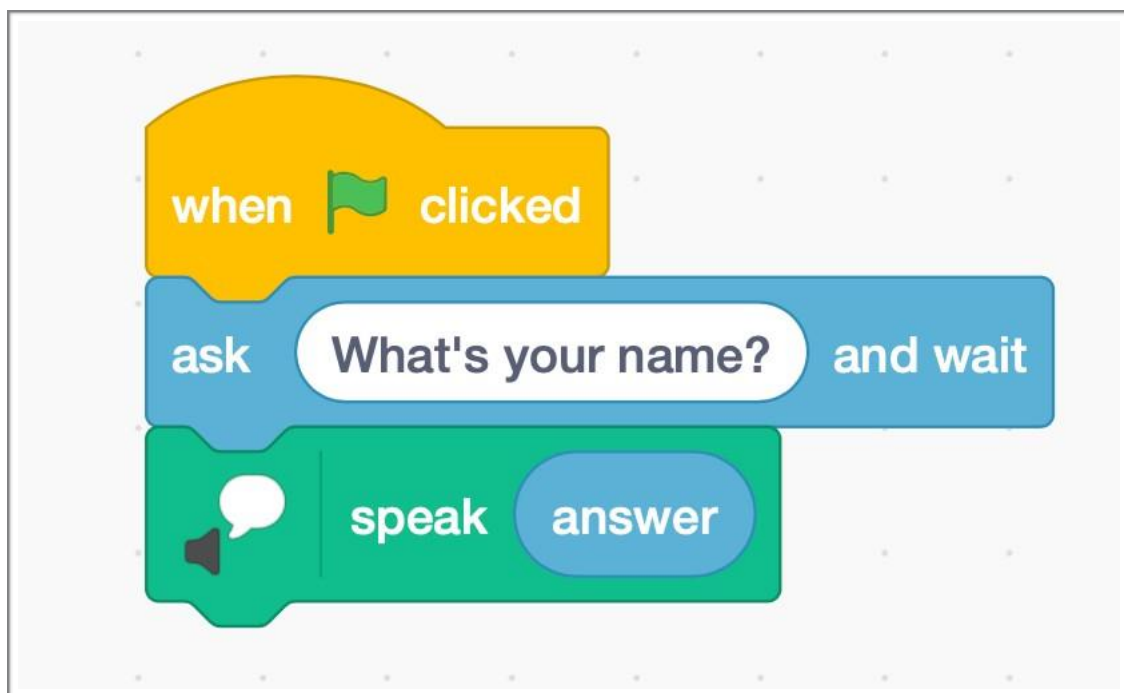


Ce este *Text to Speech*?

Acest bloc permite calculatorului să citească ceea ce s-a scris în formă textuală, pentru a face interacțiunea mai interesantă.



Următorul program permite solicitarea numelui și introducerea acestuia de la tastatură. Folosind *Text to Speech* programul poate citi numele. Programul se poate îmbunătăți solicitând diferite informații și furnizând diferite răspunsuri.



micro:bit



Scratch cu micro:bit

Micro:bit e un microcontroler de buzunar proiectat să ajute copii să învețe să codeze și să creeze cu ajutorul tehnologiei. El are numeroase caracteristici, inclusiv un display cu LED-uri, butoane și senzor de mișcare. Poate fi conectat și programat cu Scratch, MakeCode și Python. BBC micro:bit conține un accelerometru ce poate detecta dacă e scuturat sau în ce direcție este ținut micro:bit-ul. Accelerometrul poate detecta și câteva forțe ce acționează asupra acestuia. Micro:bit are un senzor de temperatură integrat ce poate detecta temperatura curentă a dispozitivului, în grade Celsius. Lista completă a caracteristicilor e prezentată în tabelul următor:

Caracteristică	Descriere
2 butoane	Butoane programabile
25 LED-uri	Pot fi programate individual pentru a afișa forme, texte sau numere
Conector USB	Pentru conectarea la un calculator pentru alimentare sau pentru încărcarea programelor pe micro:bit
Accelerometru	Detectează dacă micro:bit-ul e deplasat, înclinat, scuturat sau în cădere liberă și cu ce accelerație
Compas	Detectează în ce direcție e îndreptat micro:bit-ul
Procesor	Unde e executat programul

Radio	Comunică cu alte dispozitive micro:bit pentru jocuri multi-player
Antena Bluetooth	Trimite și primește wireless semnale către dispozitive cu Bluetooth (calculatoare, telefoane sau tablete)
Buton de resetare	Repornește micro:bit-ul
Conector pentru baterie	Permite alimentarea micro:bit-ului cu baterii
Senzor de temperatură	Detectează temperatura curentă a micro:bit-ului în grade Celsius
Senzor de lumină	LED-urile de pe micro:bit pot acționa și ca senzori de lumină pentru a detecta lumina ambientală
Pini	25 de conectori externi, numiți pini, la marginea micro:bit-ului permit conectarea altor electronice, inclusiv LED-uri, motoare și alți senzori. Aceștia se pot comporta ca intrări sau ieșiri.

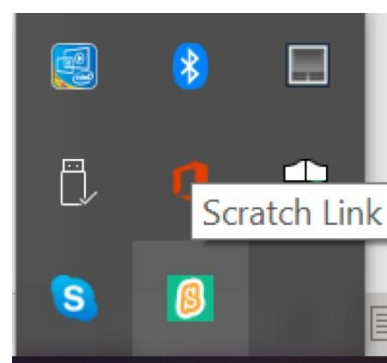
Pentru moment, Scratch 3.0 nu include blocuri care să permită utilizarea termometrului, compasului sau senzorului de lumină. De asemenea pot fi folosiți doar 3 din cei 25 de pini.

Cum se folosește

Micro:bit poate fi programat atât pe sisteme desktop (Macs, PC-uri, Chromebooks, Linux, inclusiv Raspberry Pi) cât și sisteme mobile. Pentru a putea utiliza micro:bit cu Scratch, trebuie urmați pașii prezentați în continuare (pentru Windows).

Pasul 1: Instalați Scratch Link din magazinul Microsoft sau folosiți legăturile din pagina <https://scratch.mit.edu/microbit>

Pasul 2: Porniți Scratch Link și asigurați-vă că funcționează. Ar trebui să apară în bara de instrumente conform imaginii.



Pasul 3: Conectați micro:bit-ul la calculator folosind cablul USB. Micro:bit-ul va apărea pe calculator ca un drive cu numele 'MICROBIT'.

Pasul 4: Descărcați pe la adresa <https://scratch.mit.edu/microbit> fișierul .HEX pentru micro:bit.

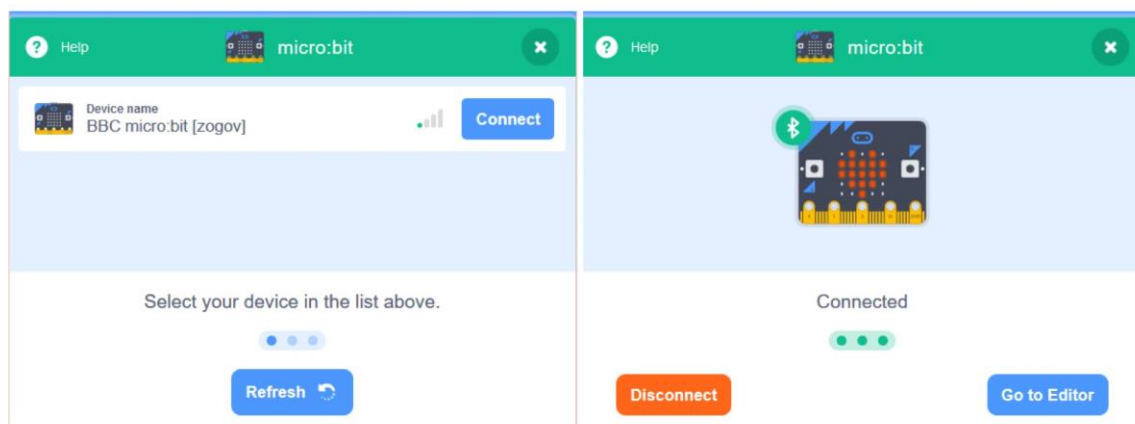
Pasul 5: Deschideți arhiva și copiați fișierul .HEX pe micro:bit

Pasul 6: Alimentați micro:bit-ul prin USB de la calculator sau cu baterii.

Pasul 7: Folosiți editorul Scratch

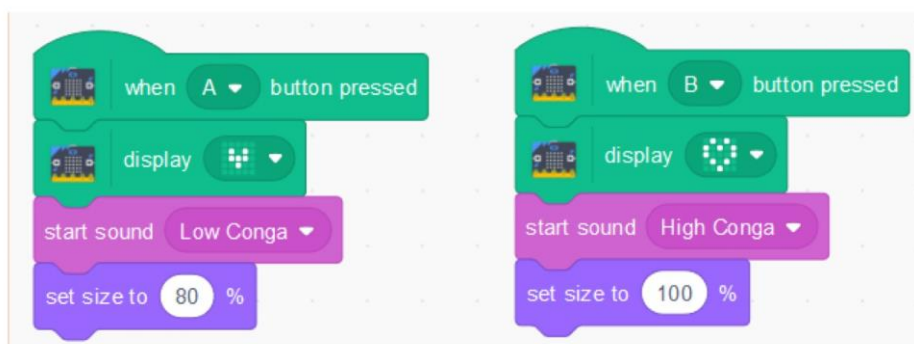
Pasul 8: Adăugați extensia micro:bit

Pasul 9: Apăsați *Connect* și *Go*



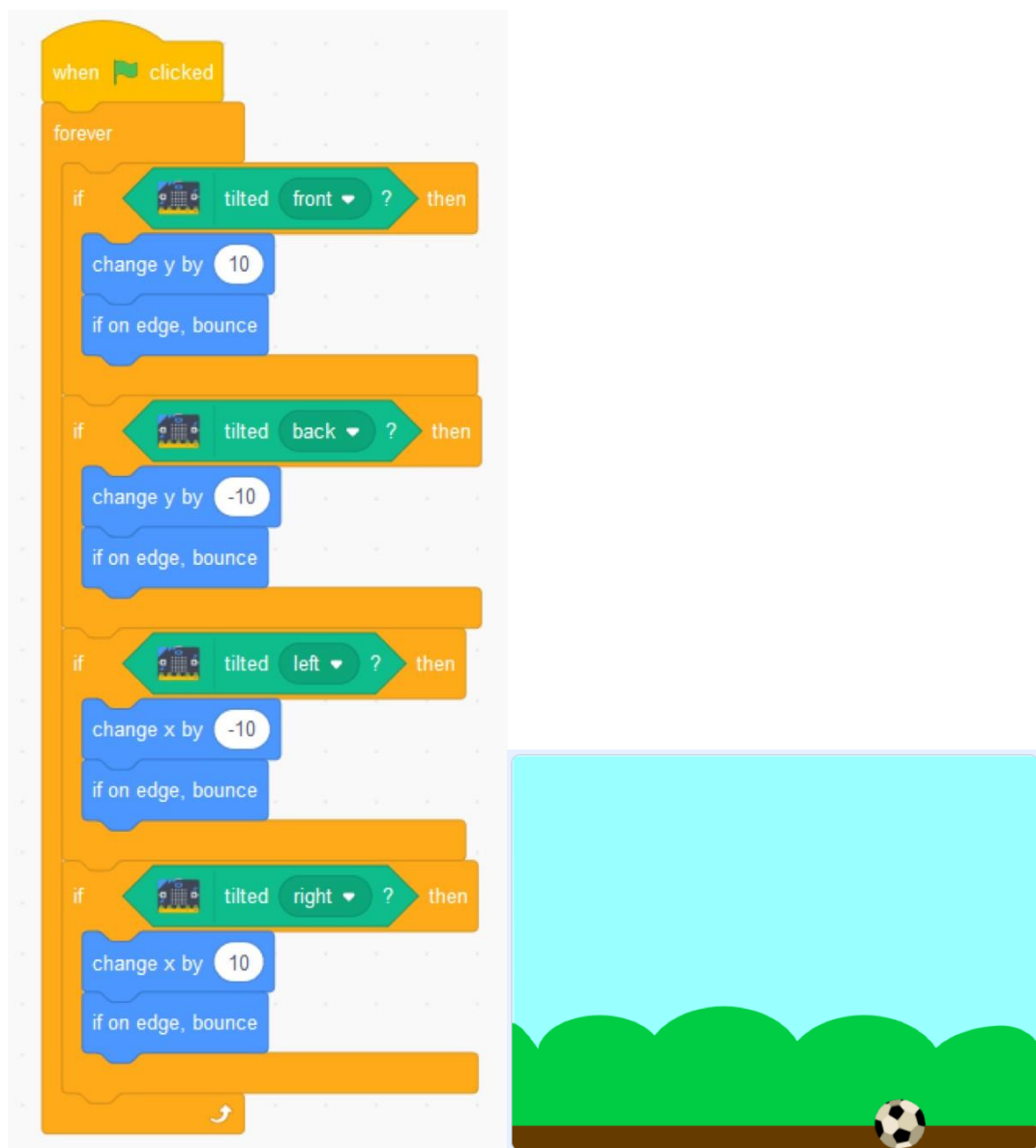
Provocarea 1: O inimă bătând

Aplicația (de la adresa <https://scratch.mit.edu/microbit>) e controlată de cele două butoane ale micro:bit-ului. În funcție de butonul apăsător se afișează pe display cele două inimi (una mică și una mare) și, în același timp, în Scratch se afișează o inimă ce își schimbă dimensiunea și se aud două sunete diferite.



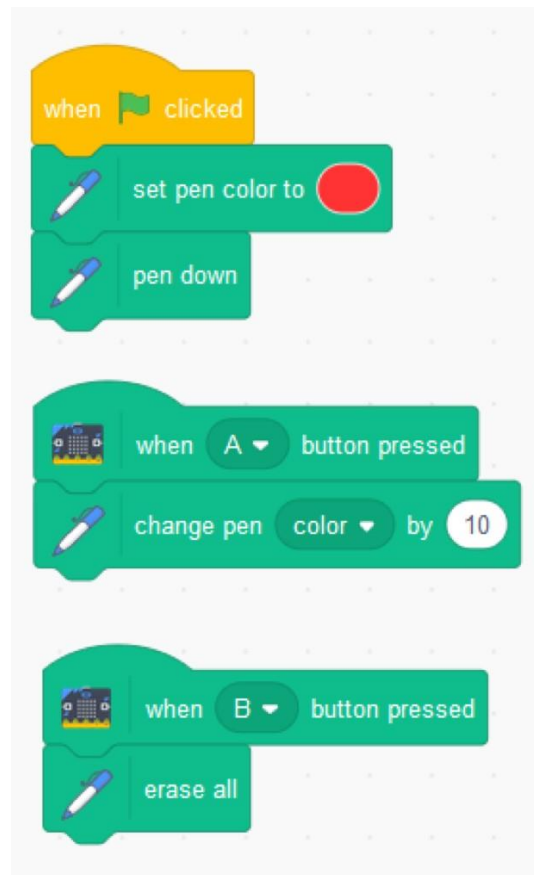
Provocarea 2: Minge

În funcție de modul de înclinare a micro:bit-ului, o minge poate fi deplasată pe ecran.

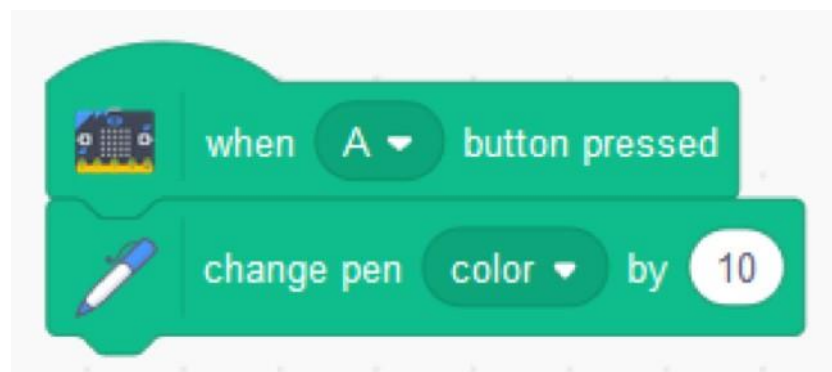


Provocarea 3: Stilou

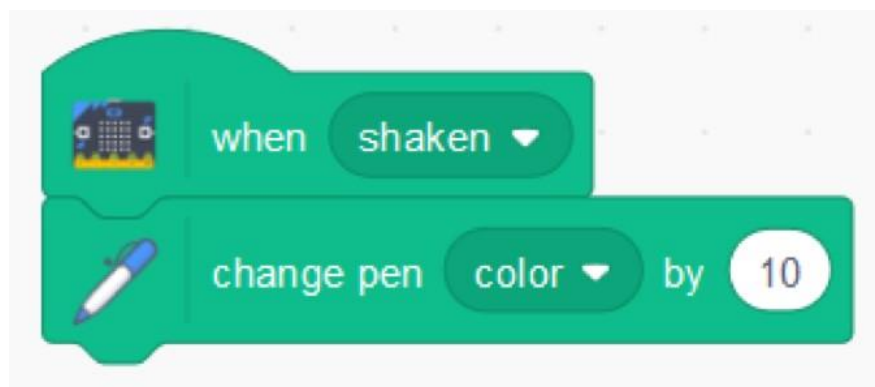
Adăugați extensia *Pen* (ro. „*Stilou*”). La aplicația anterioară, adăugați următoarele scripturi. Testați rezultatul obținut.



Dacă înlocuiți



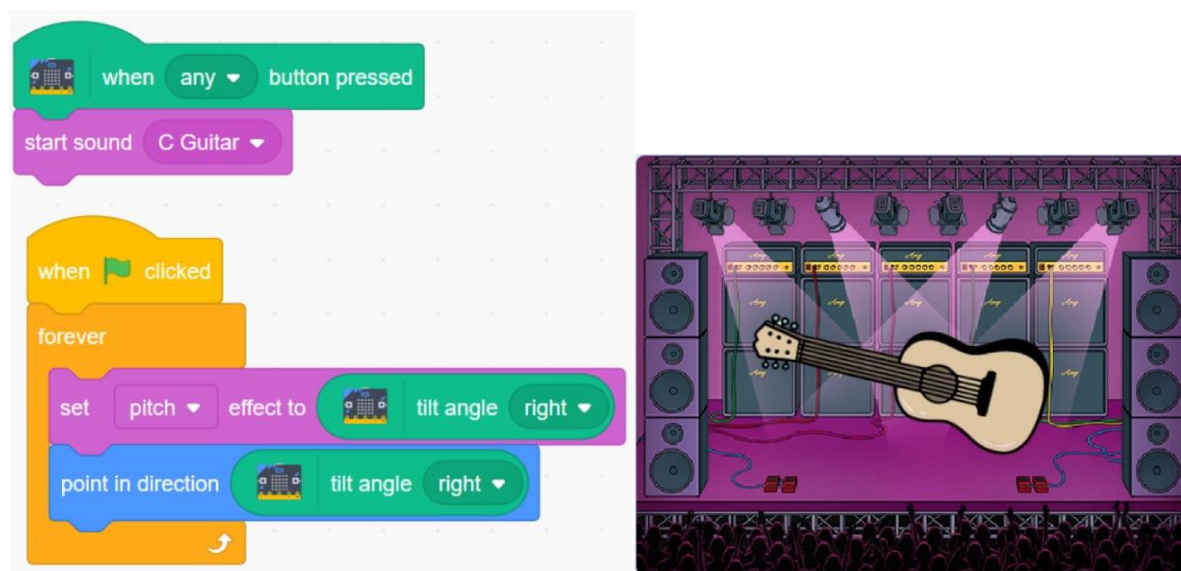
cu



atunci culoarea se va schimba atunci când micro:bit-ul este scuturat.

Provocarea 4: Chitara

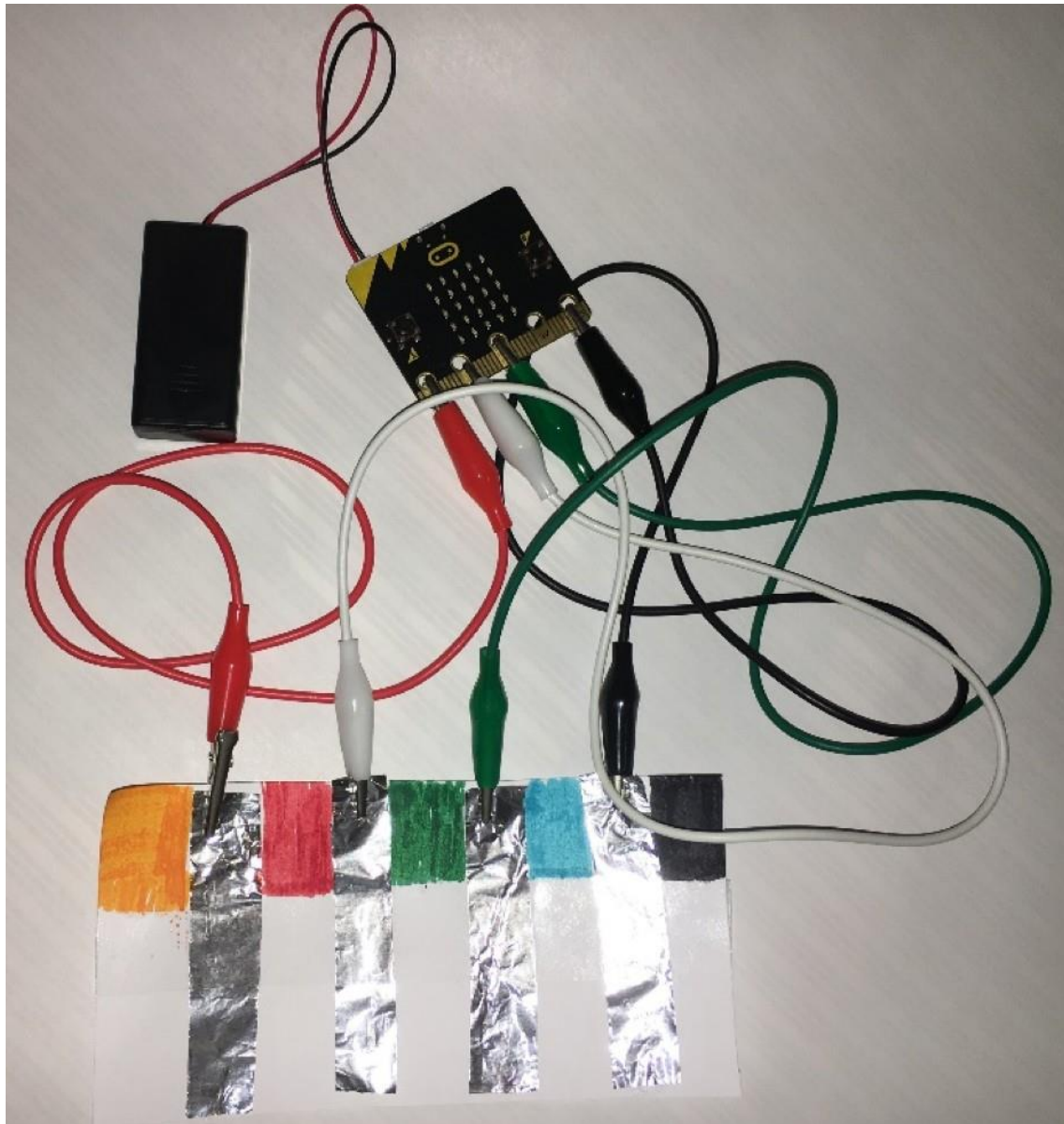
Aplicația cântă un sunet atunci când se apasă oricare buton al micro:bit-ului cu un efect, în funcție de gradul de înclinare a micro:bit-ului.



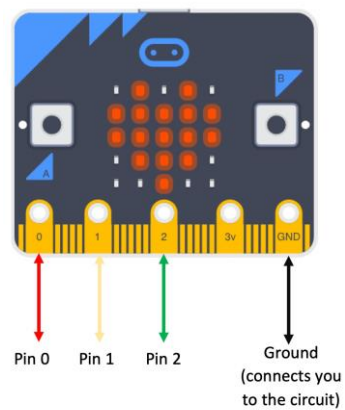
Provocare 5: Pian

Pentru a construi următoarea aplicație (inspirată dintr-un exemplu de la adresa <https://microbit.org/scratch/>) veți avea nevoie de un micro:bit, 4 cabluri cu clești, folie de aluminiu, foarfece, lipici și hârtie.

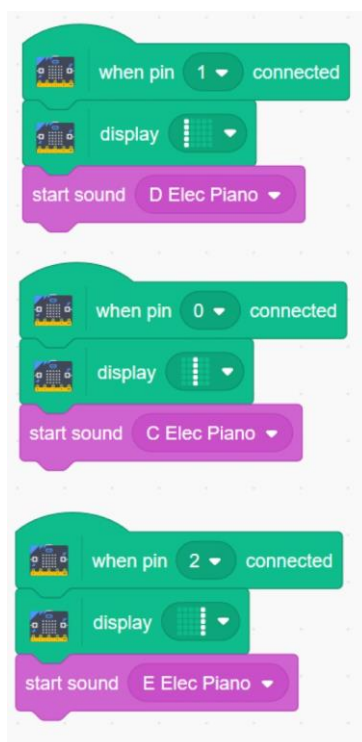
Construiți un pian asemănător celui din imaginea următoare. Conectați fiecare pin (pinii 0 – 2 și masa) la câte o bandă de aluminiu.



Schema pianului e prezentată în imaginea următoare.



Scrieți scriptul:



Testați aplicația apăsând clapele pianului (benzile de aluminiu): un deget trebuie să atingă tot timpul masa (en. *ground*) pentru a închide circuitul.

Mai mult provocări

1. Construiți un joc “piatră, hârtie, foarfece”. Aplicația ar trebui să afișeze în mod aleatoriu o piatră, o hârtie sau un foarfece atunci când micro:bit-ul e scuturat.
2. Creați propriul dispozitiv ce poate fi purtat, cu o interfață în Scratch. Creați o carcasă și tipăriți-o cu o imprimantă 3D.

MakeCode for micro:bit

Microsoft MakeCode e un mediu bazat pe web pentru învățarea programării unor dispozitive fizice de calcul, precum micro:bit-ul. MakeCode e gratuit și funcționează pe toate platformele și navigatoarele web.

MakeCode e disponibil online ca editoare online pentru:

- 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/>

și ca aplicație desktop pentru:

- micro:bit
- Adafruit
- Cue by Wonder Workshop


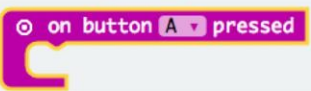


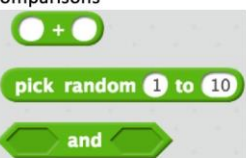
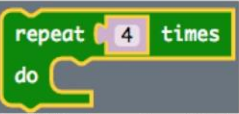
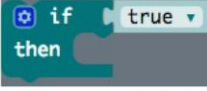


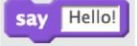

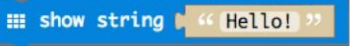

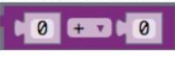
Aplicația desktop MakeCode for micro:bit are câteva caracteristici în plus față de versiunea online. Cu aplicația desktop se poate programa direct prin USB, fără necesitatea copierii fișierului HEX pe drive-ul MICROBIT, și citește serial datele de la micro:bit.

Scratch vs MakeCode for micro:bit

Pe pagina de suport pentru micro:bit:

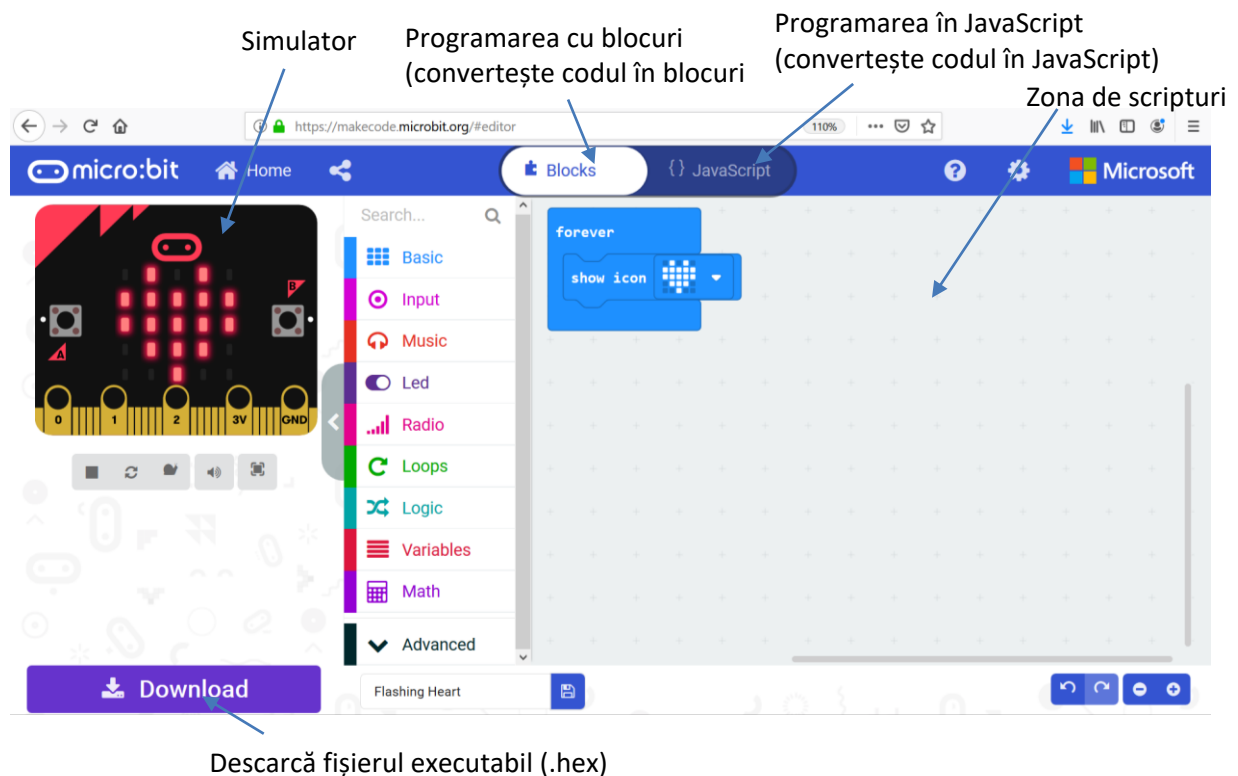
<https://support.microbit.org/support/solutions/articles/19000080171-moving-between-scratch-and-makecode> e publicat un tabel comparativ, prezentat mai jos, între Scratch și Makecode for micro:bit.

Acesta evidențiază diferențele subtile dintre câteva dintre blocurile importante. Tabelul ar putea fi util pentru utilizatorii uneia dintre cele două editoare, pasionați de micro:bit, pentru a începe utilizarea celui alt editor.

Scratch Events wait for a user action, like clicking the green flag in scratch or pressing a button on the keyboard 	Makecode 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 it's own loop (it's triggered as soon as the micro:bit is powered on). 
Operators let you do arithmetic and make comparisons 	Repeats are found in the Loops menu of Javascript Blocks  And if blocks are found in the Logic menu 
Data lets you define variables that might change within your program. Here we have made a variable called item 	Arithmetic and random number pickers can be found in Math , whereas a comparison between something and something else is found in Logic 
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 	We can define these in the Variables menu of makecode  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 

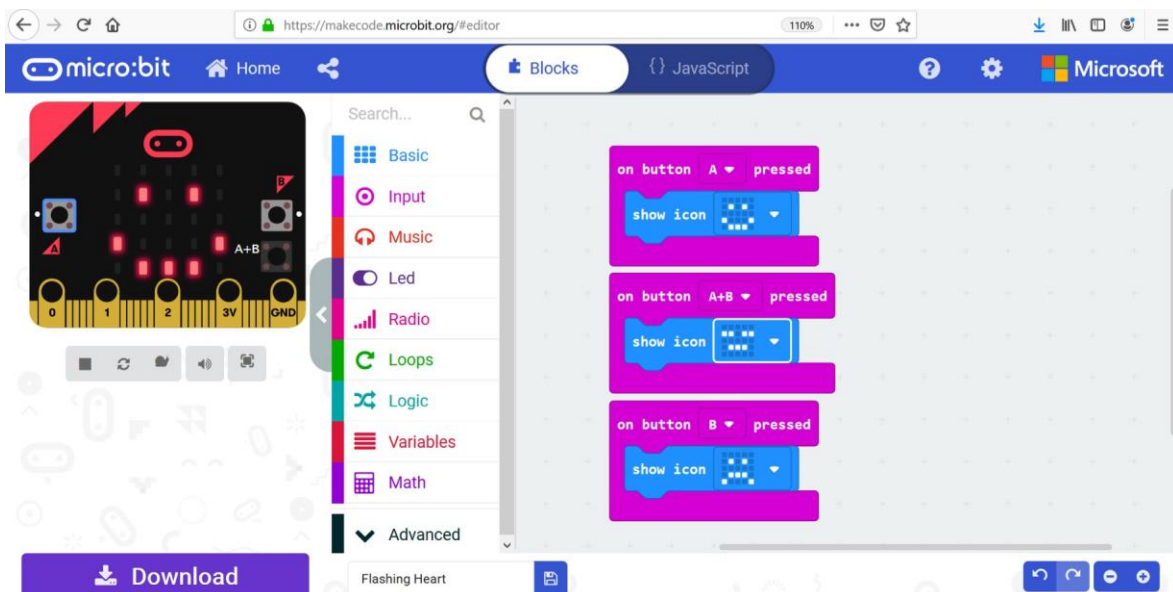
Un mare avantaj al Scratch-ului, comparativ cu varianta online Makecode, este faptul că permite programarea OTA (Over-the-Air), prin comunicare Bluetooth. Marele dezavantaj este că Scratch-ul oferă doar un set de blocuri de bază pentru micro:bit (doar 10). Pentru moment, setul mai bogat de blocuri, face Makecode un editor mai bun.

Interfața



Provocarea 1: Fețe

Această aplicație afișează o față zâmbitoare când se apasă butonul A, o față tristă când se apasă butonul B și o față adormită când sunt apăseate ambele butoane (A și B).



În zona de simulare sunt reprezentate butoanele A, B și A+B ce pot fi utilizate pentru a simula modul în care va funcționa micro:bit-ul la apăsarea acestor

butoane.

Pentru a pune aplicația pe micro:bit trebuie descărcat scriptul (ca fișier .hex). Se conectează micro:bit-ul la calculator cu un cablu USB. Se localizează fișierul .hex și se copiază pe drive-ul MICROBIT (se poate folosi tehnica drag-and-drop).

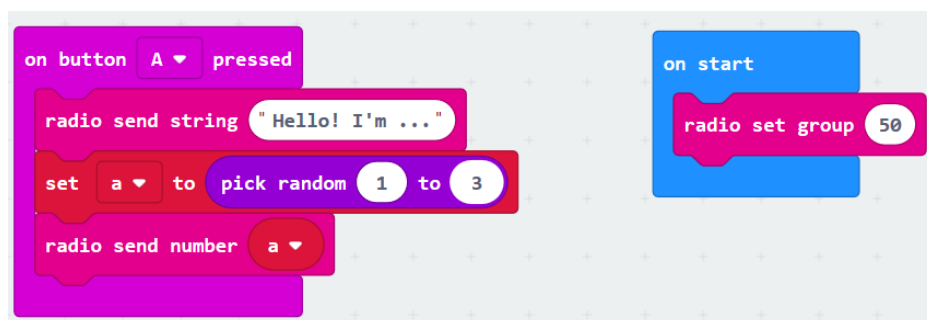
Această aplicație este inspirată din proiectele de la adresa:

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

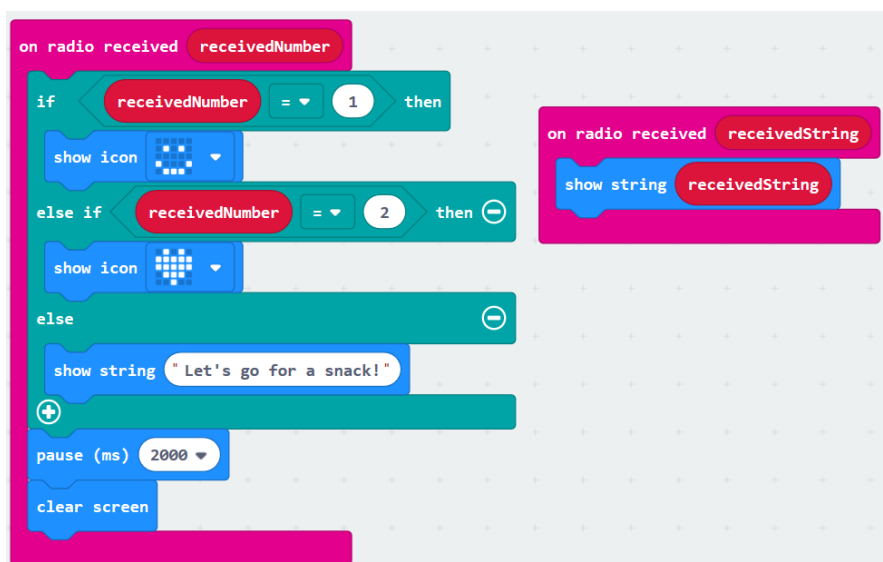
Testați și alte proiecte!

Provocarea 2: Chat

Următoarea aplicație permite două sau mai multe dispozitive micro:bit să comunice unele cu altele prin conexiune radio. Variabilele *receivedString* și *receivedNumber* sunt “trase” din blocul *on radio received*. Aplicația trebuie copiată pe fiecare dispozitiv micro:bit.



Mesajele sunt trimise doar dispozitivelor micro:bit din același grup



Mai multe exemple ce folosesc conexiunea radio pot fi găsite la următoarele adrese:

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

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

Provocarea 3: Muzică și lumină

Următoarea aplicație permite crearea de muzică prin varierea intensității luminii pe senzorii de lumină ai micro:bit-ului.

Materiale necesare:

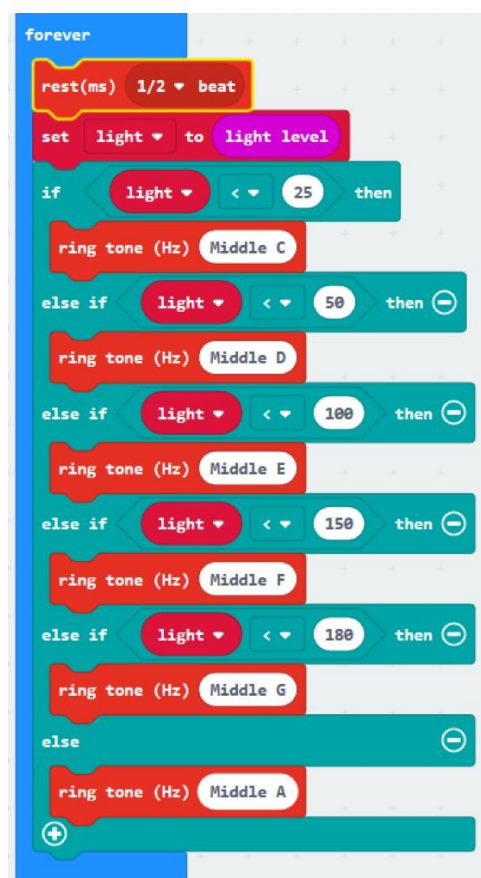
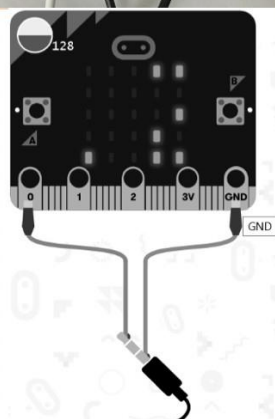
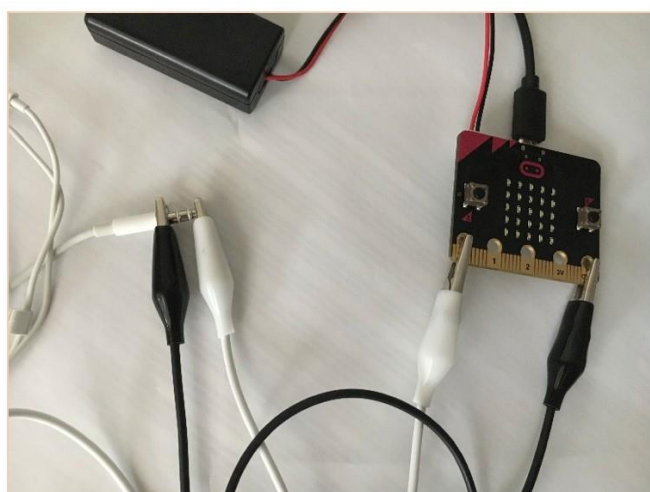
1x Micro:bit

1x căști audio

2x cabluri cu clești

1x cablu USB

Pentru început, trebuie conectate căștile audio la micro:bit conform următoarelor imagini:



Se definește o nouă variabilă cu numele *light* (din secțiunea *Variables*) și se scrie

codul. Se copiază fișierul .hex pe drive-ul MICROBIT. Nivelul de lumină e o valoare între 0 (întuneric) și 255 (lumina maximă). Intensitatea luminii e măsurată de LED-urile ce formează display-ul micro:bit.

Provocarea 4: Muzică și... fructe și legume

Aceasta este o aplicație amuzantă ce folosește fructe și legume pentru a crea muzică prin închiderea unui circuit.

Materiale necesare:

1x micro:bit

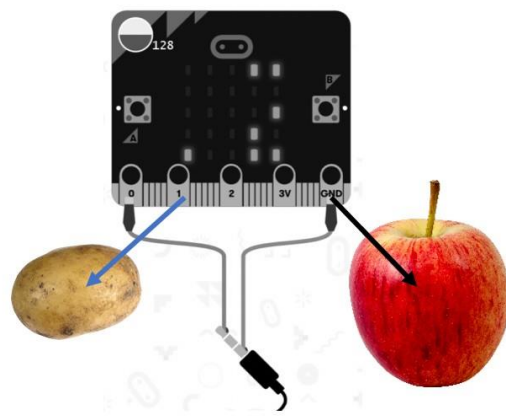
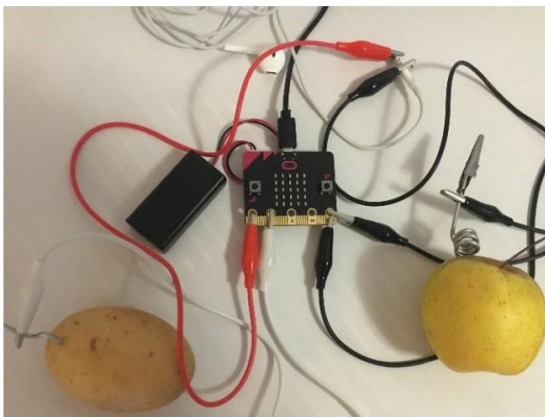
1x căști audio

4x cabluri cu clești

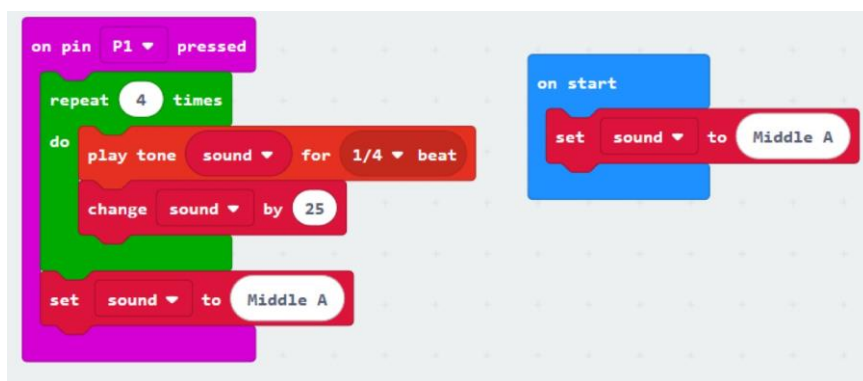
1x cablu USB

2x fructe sau legume (cartofi, banane, portocale, mere, etc...)

Realizați următoarele conexiuni:



Creați o variabilă cu numele *sound* pentru a memora o notă muzicală. Scrieți următorul cod:



Salvați fișierul .hex și copiați-l pe dive-ul MICROBIT. Creați muzică ținând fructul/leguma conectată la masă (mărul în exemplul nostru) și atingând fructul/leguma conectat/ă la pinul 1.

Puteți conecta încă un fruct/legumă la pinul 2 pentru a crea și alte sunete. În acest caz copiați codul corespunzător pinului 1, selectați pinul 2 și modificați spre exemplu valoarea 25 cu -25.

Provocarea 5: Servomotor

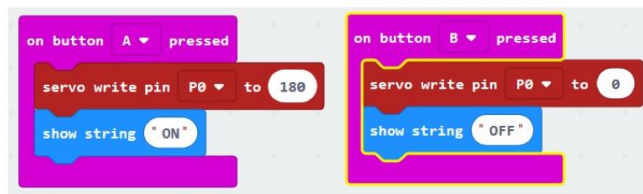
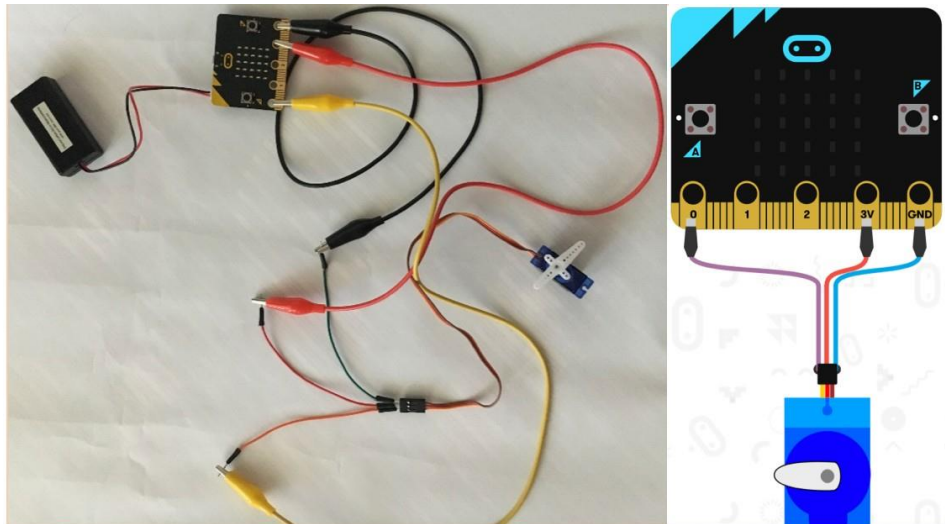
Această aplicație arată cum puteți conecta un servomotor la un dispozitiv micro:bit.

Materiale necesare:

- 1x micro:bit
- 3x cabluri cu clești
- 1x cablu USB
- 3x fire Dupont tată-tată
- 1x servomotor TowerPro SG90

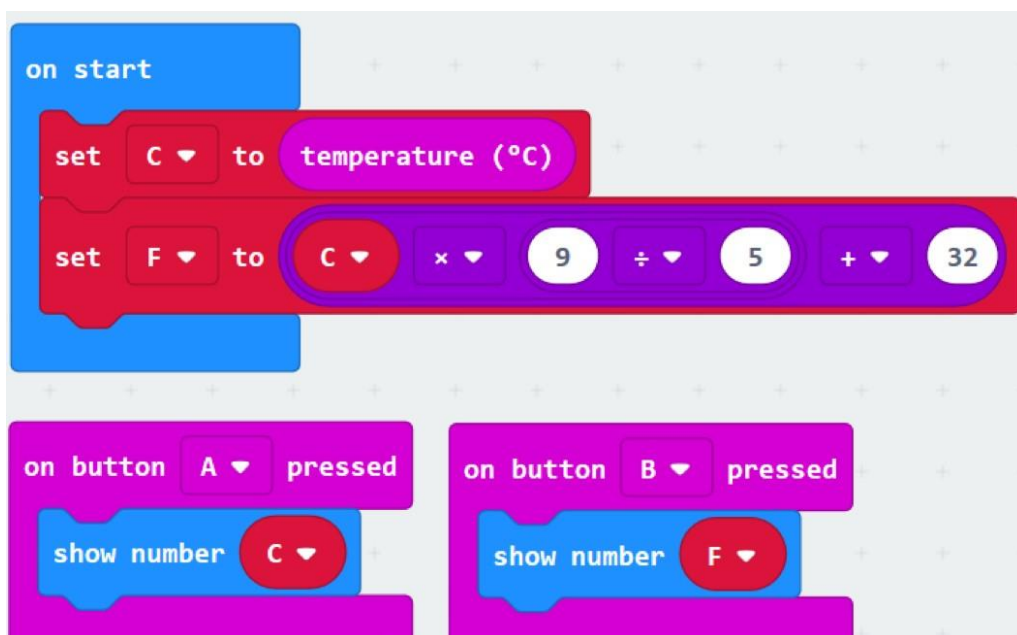
Servomotorul poate fi conectat după cum urmează:

micro:bit	servomotor
GND	Fir maro
3V	Fir roșu
P0	Fir portocaliu



Provocare 6: Termometru

Această aplicație folosește senzorul de temperatură integrat pentru a afișa temperatura în grade Celsius, atunci când se apasă butonul A și în grade Fahrenheit când se apasă butonul B.



Pentru a obține o valoare mai apropiată de cea reală, trebuie comparată valoarea afișată de micro:bit cu cea obținută cu un termometru real. Programul poate fi modificat astfel încât să realizeze scăderea diferenței dintre cele două valori comparate.

Provocarea 7: Busolă

Următoarea aplicație afișează punctul cardinal înspre care e îndreptat micro:bit-ul. După scrierea codului următor și copierea fișierului .hex pe micro:bit, dispozitivul va solicita calibrarea. Pentru a realiza calibrarea, dispozitivul trebuie aplecat în toate direcțiile până se obține aprinderea tuturor LED-urilor. Dacă s-a realizat cu succes calibrarea, se va afișa o față zâmbitoare.



Mai multe provocări

1. Realizați un numărător de pași.
2. Creați-vă propriile aplicații având în vedere materiile pe care le predăți!

Resurse

<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>

Referințe

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

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

“Fie că dorești să dezvălui secretele universului sau doar să-ți faci o carieră în secolul 21, învățarea programării de bază e esențială.”

- *Stephen Hawking, fizician teoretician și cosmolog*