

Explore STEM & Coding with

EDU:BIT

on start

start melody power up ▾ repeating once ▾

set all RGB pixels to 

forever

if IR sensor triggered then

Set servo S1 ▾ position to 40 degrees

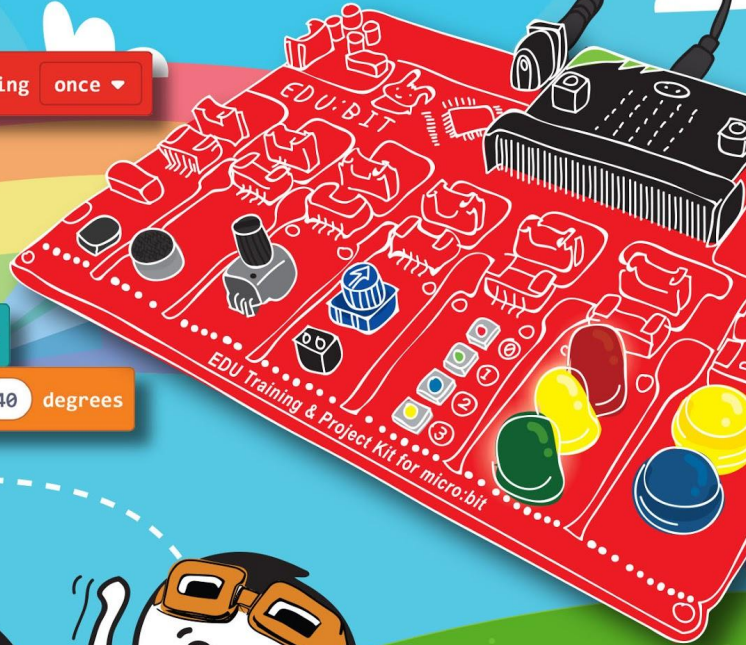
show leds



else

Set servo S1 ▾ position to 20 degrees

show arrow East



Kedves _____

Hallottál már a micro:bit-ről? Az Egyesült Királyságban tervezett, az egész világon elterjedt programozható mikroszámítógép, amely arra ösztönzi a gyerekeket hogy szórakoztatóan és egyszerűen tanulják a programozást (kódolást).

A Cytron mérnökei kifejlesztettek egy EDU:BIT nevű szettet, amely segít a kódolás megértésében. Tartalmaz egy **Music Bit** piezo hangszórót és audio aljzatot, hogy zenét játszhass, **Sound Bit** a hang/zaj észlelésére (mérésére), **Potentio Bit** az analóg vezérléshez, **IR Bit** az objektum felismeréshez, **RGB Bit** a színes fény kijelzéshez, **Traffic Light Bit** a jelzőlámpa fényei piros, sárga és a zöld LEDEK, és végül a **Button Bit**, nyomógombok (kapcsolók). Ezzel a készlettel kapsz egy **DC** és egy **szervomotort** is, amelyekkel játszhatsz.

Hogy ez mennyire menő?

Szerinted elkezdhetjük? A következő oldalakon néhány klasszikus gyermekkori játékot fogunk bemutatni, illetve létrehozni, ilyen például a Kő-Papír-Olló, Kígyók és Létrák Játéka, Twister, Simon azt mondja, stb. Kövesd a részletes utasításokat a játékok felépítéséhez, majd érezd jól magad a barátaiddal játszva. Alkosd meg a saját programjaidat az elkészített játékokhoz.

Minden fejezet végén lesz egy gyakorlás, amiben a megtanult dolgokat fogjuk használni. Ha elakadsz, mindig segítünk!

Készen állsz?

Gyerünk kezdjünk bele ebbe az izgalmas játszamába!



Hajrá!

Ádám és Anna



Fedezd fel a STEM-et és a kódolást az EDU:BIT-tel képzési és projektkészlet

Írta: Cheryl Ng, SC Lim & Adrian Teo

Fordította: Vígvári György

Rajzolta: Suhana Oazmi

2020

Kiadja a



Magyarországi forgalmazó:



VÍGVÁRI
R E N D S Z E R H Á Z



www.vigvari.hu



info@vigvari.hu

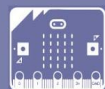


(22) 500-095

(22) 500-096



Tartalomjegyzék:



1. fejezet: Hello Világ! (Led mátrix használata)
Vágjunk bele!

1 - 11



2. fejezet: Játszunk KŐ, Papír, Olló játékot
Változók és eseményalapú programozás

12 - 25



3. fejezet: Hallgassunk zenét (Music Bit)
Programozási funkciók

26 - 38



4. fejezet: Közlekedési lámpa (Traffic Light Bit)
Digitális kimenet

39 - 47



5. fejezet: Az IR digitális kocka, gyerünk (IR bit)
digitális bemenet, tömbök és ciklusok

48 - 60



6. fejezet: Hé, itt vagy! (Potentio Bit)
Analóg bemenet, feltételes programozás

61 - 74



7. fejezet: Hallod a tapsot! (Sound Bit)
váltás a módok között a programban

75 - 87



8. fejezet: Tegyük egy kört. (DC motor)
a motor forgásiránya és sebességszabályozása

88 - 95



9. fejezet: Vigyázz-Kész...Góóól (Servo motor)
Szervomotor szöghelyzet vezérlése

96 - 104



10. fejezet: Mester, fel tudod törni a kódot?
(RGB bit) *RGB színek*

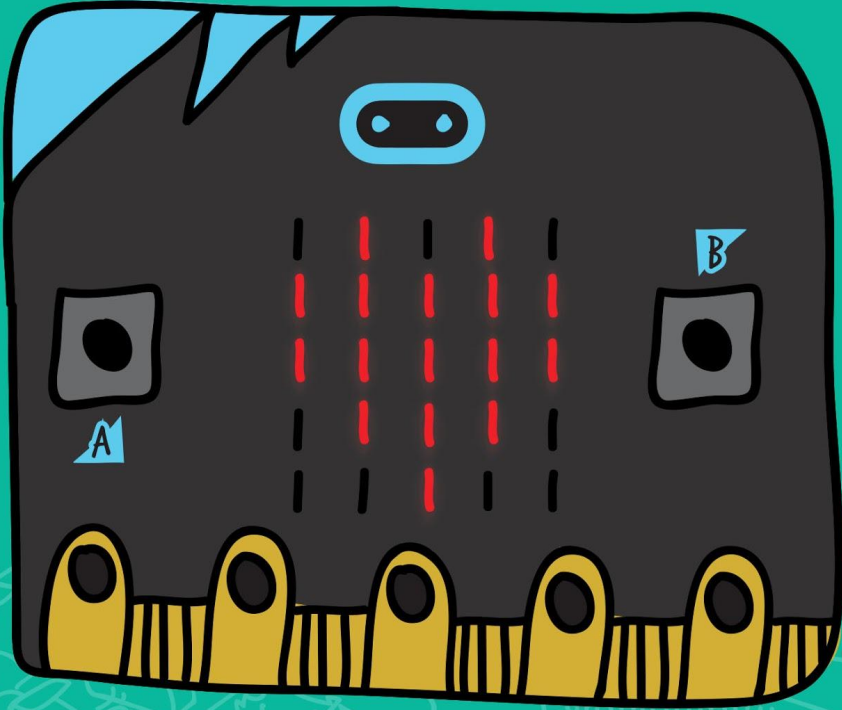
105 - 113



Bonusz fejezet: Simon mondja ledekkel,
Rádiós kommunikáció

114 - 124

> Helló, Világ!_



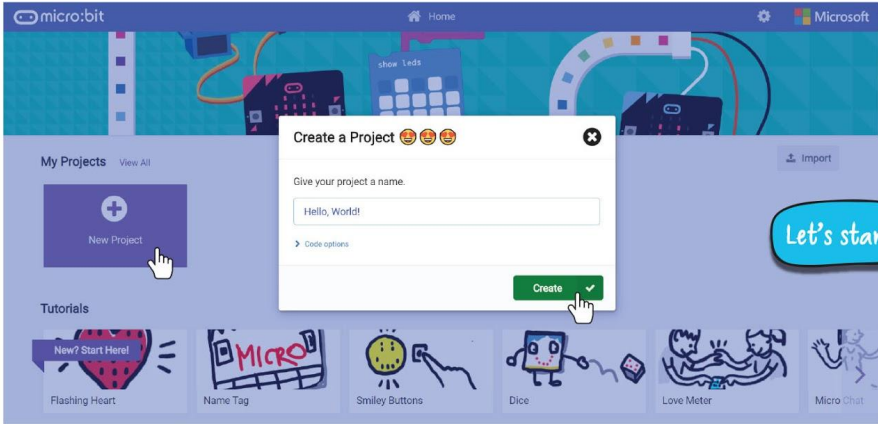
Itt is megnézheted!

link.cytron.io/edubit-chapter-1

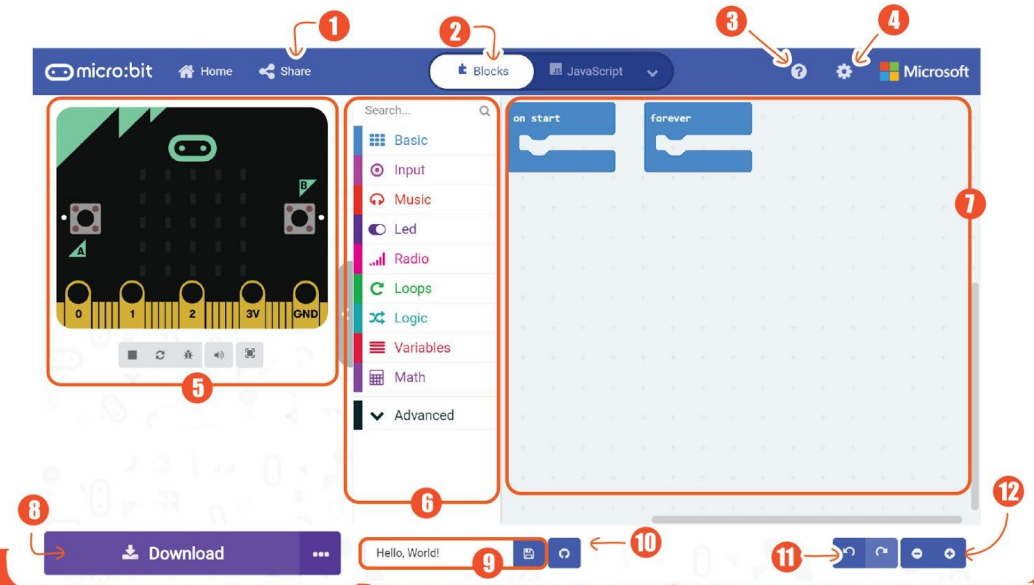
Kódolásra fel!

Első

lépés, nyissa meg a böngészőjét, majd írja be a <https://makecode.microbit.org/> Klikkeljen a „new project”, majd adjon nevet az új projektjének, majd create.



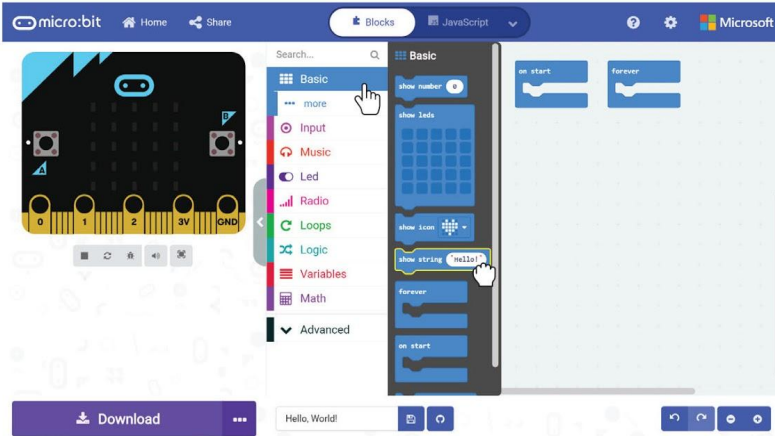
Itt található a Microsoft MAKECODE szerkesztő oldal, amely lehetővé teszi, az EDU:BIT egyszerű programozását.



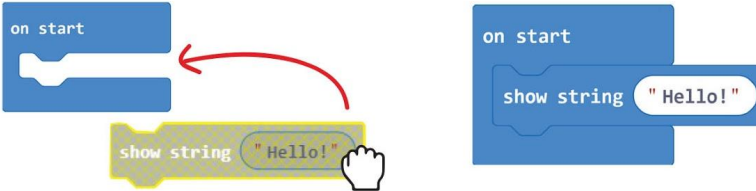
1. Közzéteszi és megoszthatja projektjét.
2. Válasszon program nyelvet
3. Nyissa meg a segítség menüjét projektet.
4. Módosítsa a beállításokat, adjon hozzá
5. Szimulátor
6. Eszköztár: .

7. Programozási Munkaterület: eten.
8. Töltse le kódját a micro: bit-re.
9. Nevezze el és mentse el az aktuális projektet.
10. Hozzon létre github-lerakatot.
11. Előre/Vissza
12. Nagyítás +/-

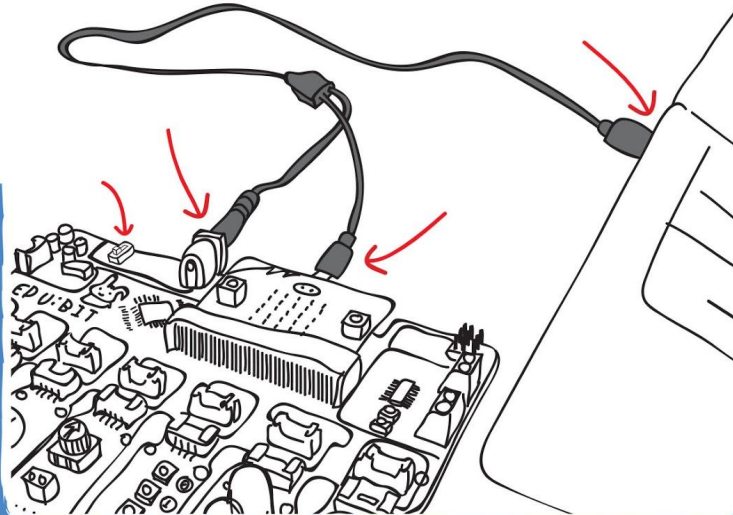
Második lépés, klikkeljen az [Basic] és válassza ki a [Show string] blokkot.



Harmadik lépés Klikkeljen és húzza be a [Show string] blokkot az indításkor blokkba.

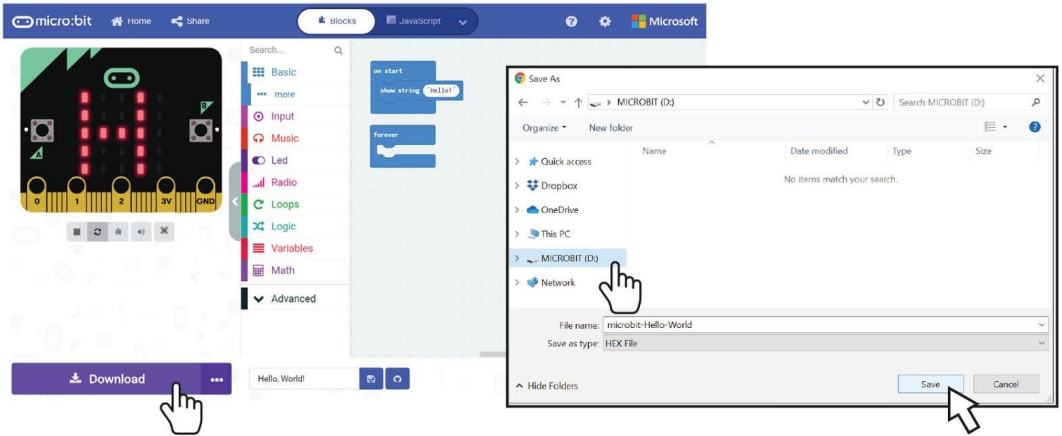


Negyedik lépés Csatlakoztassa az USB-kábelt a számítógéphez és az edu: bithez az ábra szerint. Ne felejtse el bekapcsolni az edu: bit funkciót a kapcsoló BE állásba állításával.





Ötödik lépés Kattintson a **[Download]** gombra. A felbukkanó ablakban válassza a projekt letöltését a MICROBIT meghajtóra. Csukja be az ablakot, amikor a "Letöltés kész".

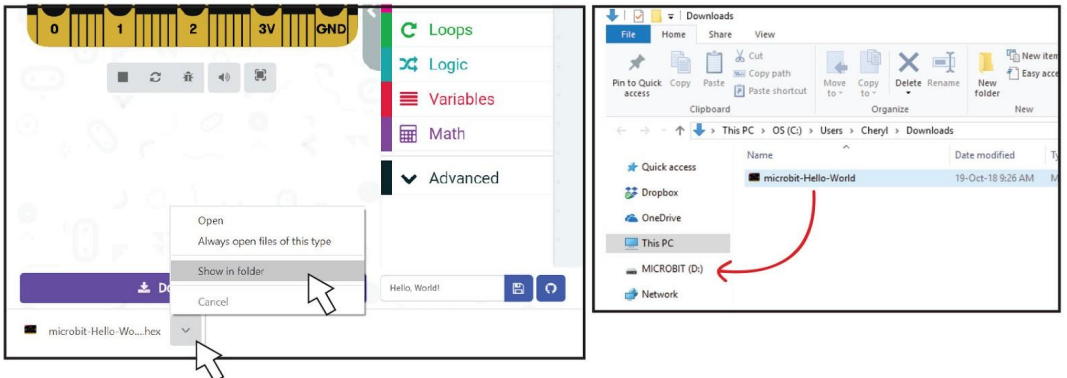


A micro:bit hátulján lévő narancssárga LED villog az átvitel során, és a befejezés után a kód automatikusan fut.

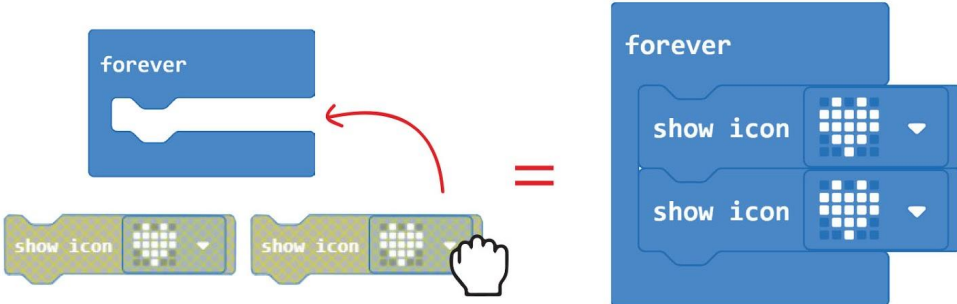


Megjegyzés!

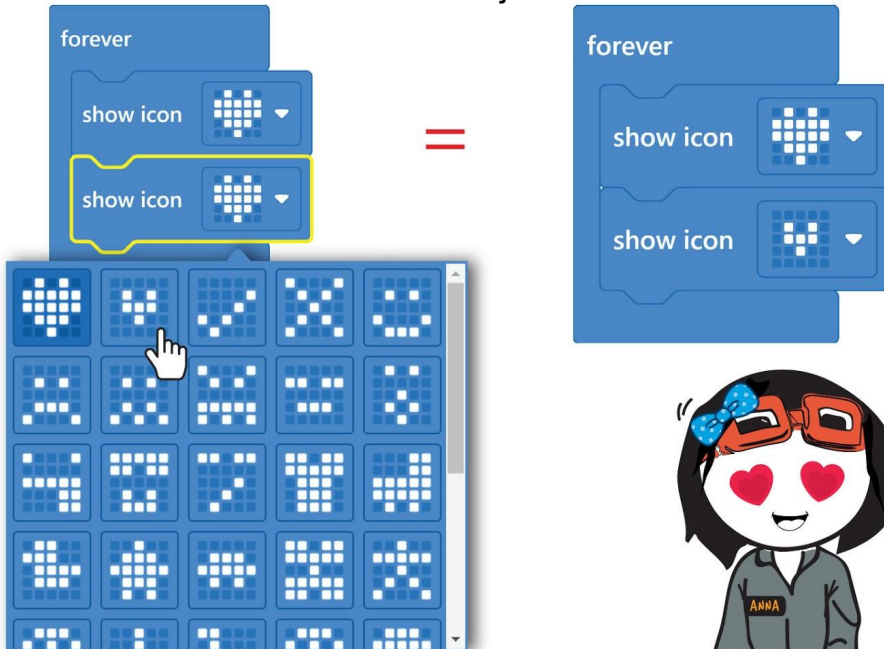
Ha a felbukkanó ablak nem jelenik meg, ez azt jelenti, hogy a fájlt automatikusan letöltötte arra a helyre, ahol a böngészője a letöltések mentésére állította be. Kattintson a jobb gombbal a letöltött .hex fájlra, amely megjelenik az ablak alján, és válassza a „Megjelenítés mappában” lehetőséget. Kattintson és húzza a letöltött "microbit-xxxx.hex" fájlt a MICROBIT meghajtóra, mintha egy fájlt másolna egy flash meghajtóra.



Hatodik lépés Kattintson az [Basic] elemre, és kattintson a [show icon] blokkjára. Ismételje meg egy további [show icon] blokk hozzáadásával. Kattintson és rögzítse az [show icon] blokkjait az [Forever] blokkban.



Hetedik lépés Bal egérgombbal kattintson a másik [show icon] blokk-ra, és a lefelé mutató nyíl segítségével válassza ki az intelligens szív kialakítását a felbukkanó ablakból. Majd töltsé le az edu:bitre



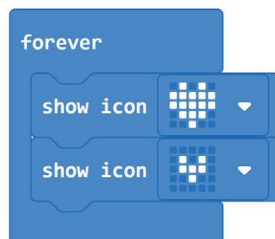
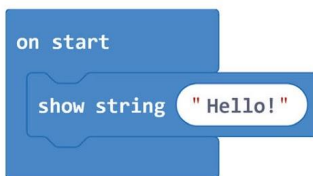
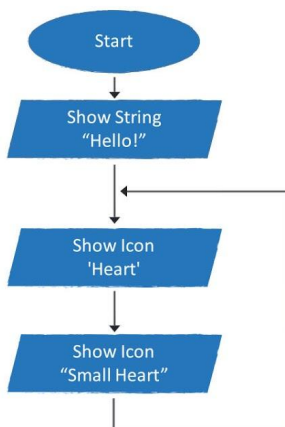
Szerelem első látásra. Látsz dobogó szív animációt?



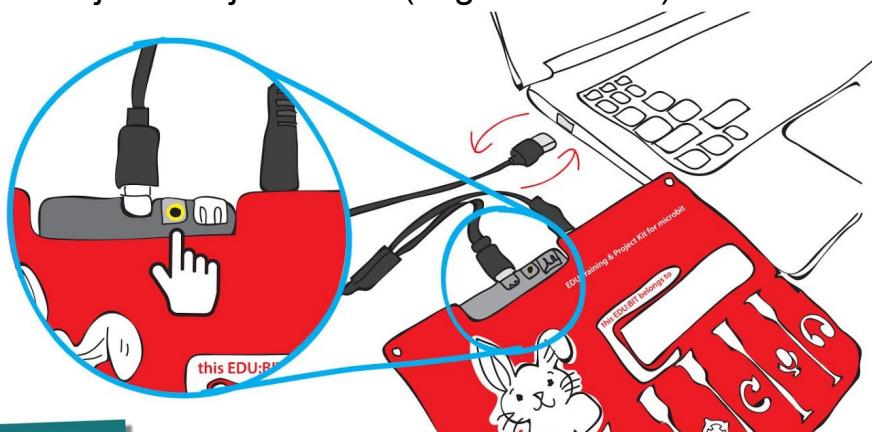
Fejtsük meg a programot!



Észrevetted, hogy a "Hello" szöveg csak egyszer gördül át a képernyőn, de a szív ikonok folyamatosan váltják egymást!



Az [on start] blokk futtatja a kódot (egyszer). Az [forever] blokk újra és újra futtatja a kódot (végtelen ciklus).

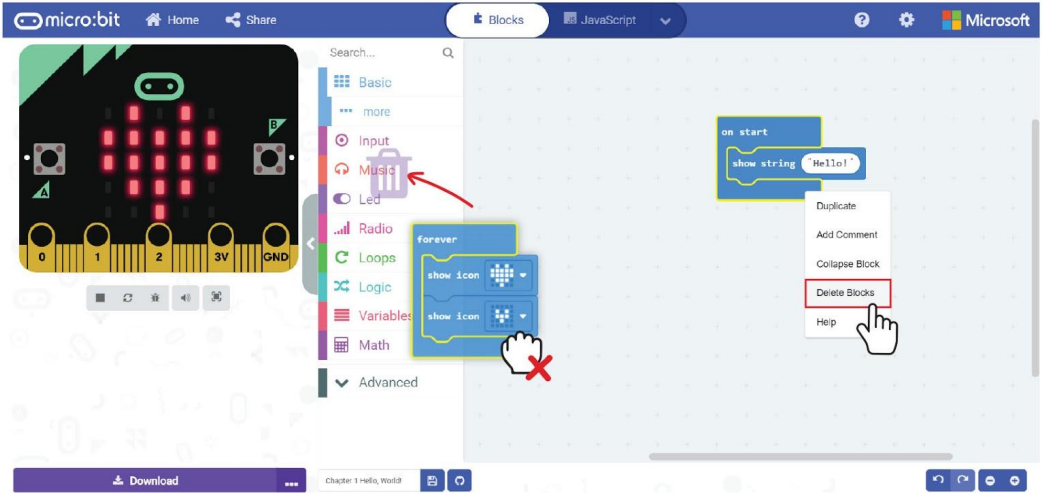


Megjegyzés!

Ha a programot újra el szeretné indítani, egyszerűen nyomja meg a RESET gombot, vagy húzza ki az USB-kábelt, majd dugja vissza ismét az alaplap visszaállításához.

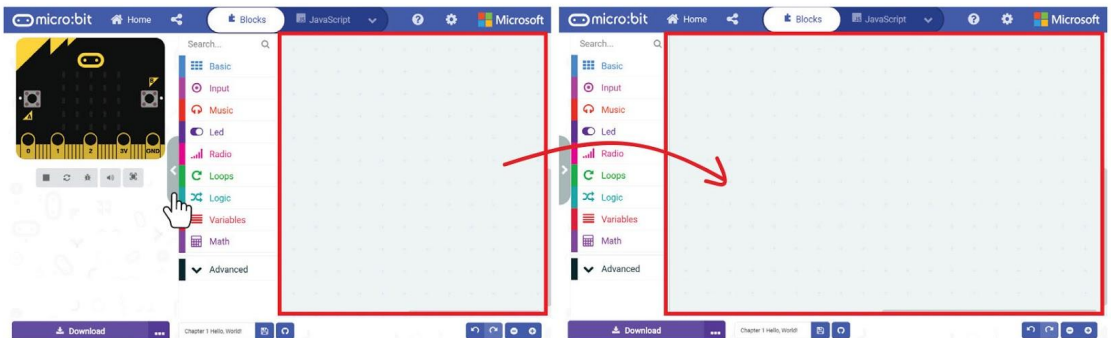
Gyors tipp # 1:

A blokk (okat) törölheti, ha rákattint és húzza a nem kívánt blokkokat az eszközkészlet területére. Engedje el, hogy törölje a blokk (oka) t, amikor a „kuka” ikon megjelenik. Alternatív megoldásként kattintson a jobb egérgombbal a blokkra, és válassza a „Blokk törlése” lehetőséget



Gyors tipp # 2:

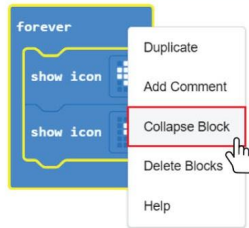
Ha nem a szimulátor ablakot használja, akkor a fülre kattintva elrejtheti azt, hogy több hely maradjon a kódolási blokkok számára





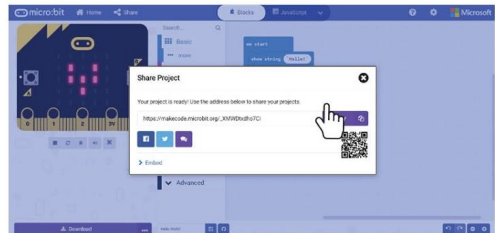
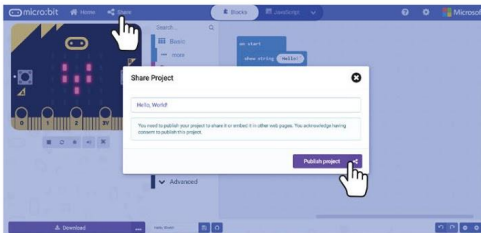
Gyors tipp # 3:

A blokk összecukásához kattintson a jobb gombbal a blokkra, majd válassza az 'Összecukás' lehetőséget. A blokkcsoport kibontásához egyszerűen kattintson az ikonra.



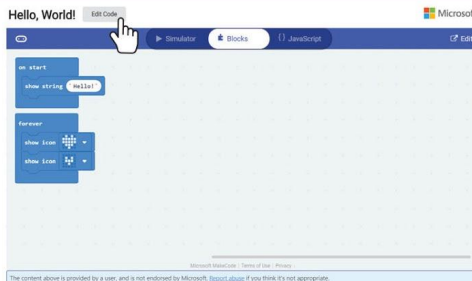
Gyors tipp # 4:

Megoszthatja a kódját tanárával vagy barátaival a projekt közzétételével, majd elküldheti számukra a projekt URL-jét. Ehhez kattintson a (Megosztás), majd a felbukkanó ablakban a (Projekt közzététele) elemre. Lát egy új felbukkanó ablakot a projekt URL-jével. Kattintson a (Másolás) gombra a projekt URL másolásához.



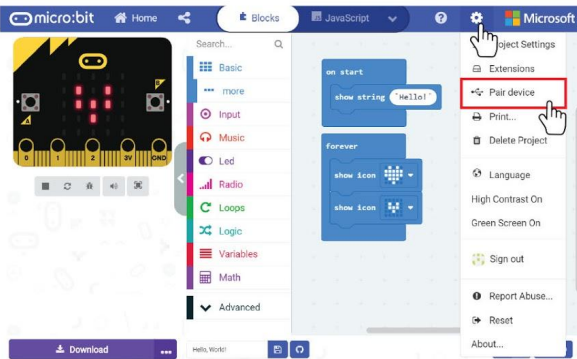
Gyors tipp # 5:

A tanár, vagy a barátai a következő oldalt fogják látni, amikor a böngészőben megnyitják a projekt URL-jét. Megtekinthetik a kódot és szerkeszthetik azt is, ha a Kód szerkesztése gombra kattintanak



Gyors tipp # 6:

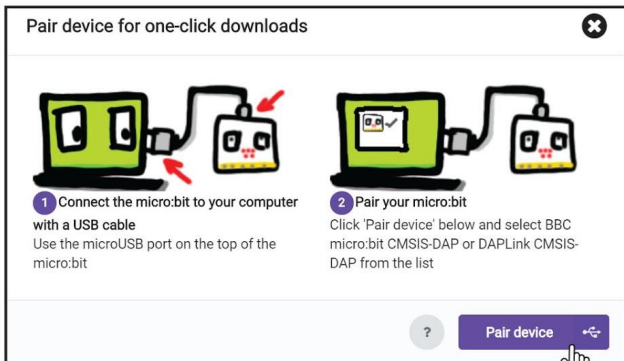
Tudja, hogy párosíthatja az eszközöket egy kattintással történő letöltésekhez? Ehhez kattintson a fogaskerékű ikonra, majd válassza az 'Eszköz párosítása' lehetőséget.



Megjegyzés!

A legfrissebb firmware-re van szüksége a micro: bit eszközön, és használnia kell az új Edge vagy a Chrome böngészőt.

Ellenőrizze, hogy az Edu: BIT csatlakoztatva van-e a számítógéphez, majd kattintson az {Eszköz párosítás} gombra, a felbukkanó ablakban. Ezután válassza ki a listából a BBC micro: bit CMSIS vagy DAPLink CMSIS-DAP elemet, majd kattintson a {Csatlakozás} gombra.



Az eszköz párosítása után a (Letöltés) gombra kattintva közvetlenül kivetítheti kódját az EDU: BIT-re. Próbáld ki!

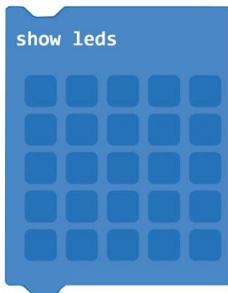
Ha problémái vannak az eszköz párosításával, akkor információt a <http://makecode.microbit.org/device/usb/webusb/troubleshoot> oldalon talál.





Fedezzünk fel további blokkokat

#1. számú elem [show leds] blokk segítségével megtervezheti a saját ikonjait, és [show number] blokkot a számok megjelenítéséhez.



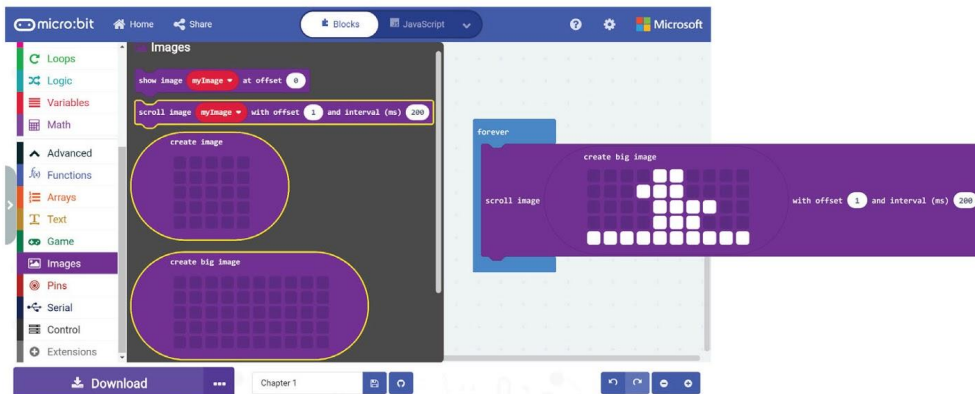
Tudtad?

1000 milliszekundum = 1 másodperc

2 Adjon hozzá egy [pause] blokkot a program lassításához. Ez a funkció szünetelteti a programot a megadott milliszekundumaként (ms).



3 Ha képet szeretne gördíteni a Led mátrix kijelzőjén, akkor a (kép létrehozása,eltolása, és intervallum(ms)) blokkokat használhatja, vagy (létrehozhat képet), vagy (nagy képet hozhat létre) blokkot a (Képek) kategóriájú fiókból (az Advanced kategória alatt)



A program futtatásakor kiskacsák láthatók, amelyek egymás után mozognak a led mátrix kijelzőn.

Alkalmazás kihívás

Az EDU:BIT program digitális hirdetőtáblaként működik.

Indításkor

Jelenítsen meg egy egyszerű animációt a figyelem felkeltésére, majd gördüljön tovább az osztály nevére

Mindig

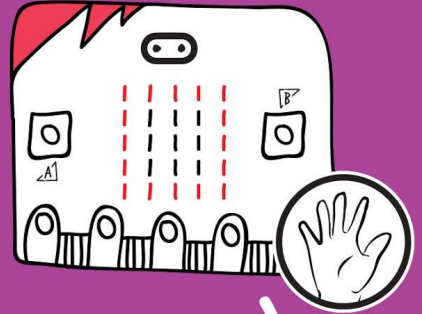
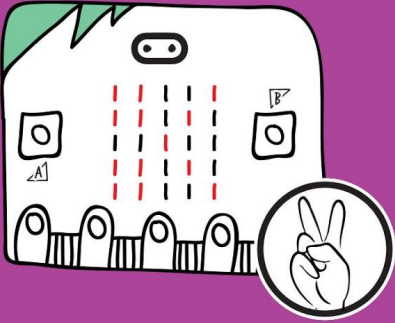
Gördüljön a mai dátumhoz és az osztályhoz kapcsolódó egyéb fontos információkhoz



Gyerünk játszunk KŐ PAPÍR OLLÓ!

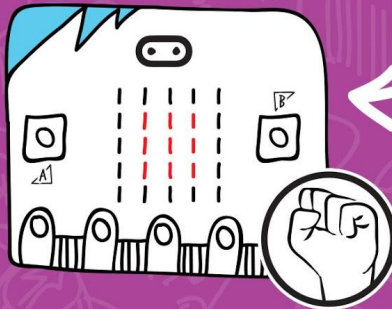
Nyomógombok a micro:biten és a Button bit

OLLÓ üti a papírt



KŐ üti az ollót

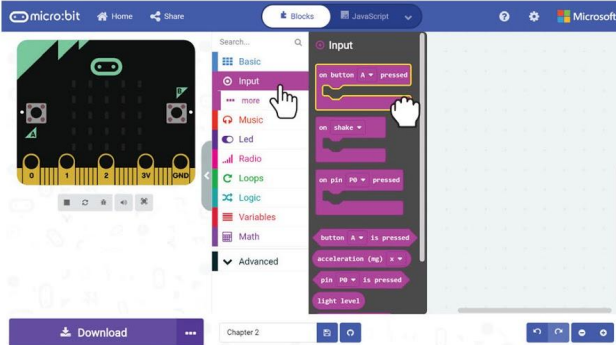
PAPÍR üti a követ



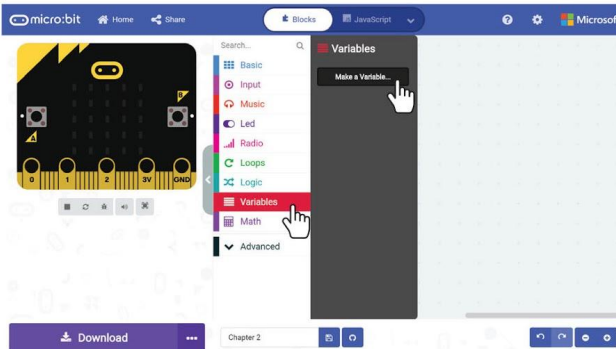
2. Fejezet kő, papír, olló

Gyerünk kódolni!

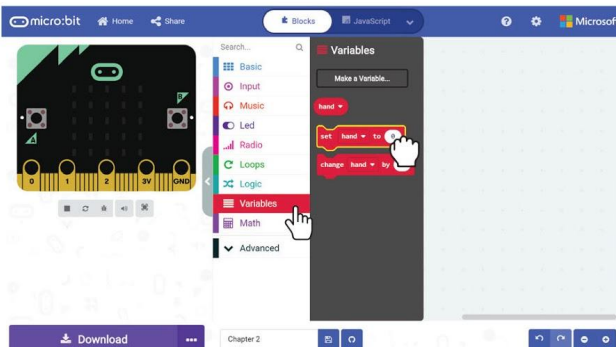
1. lépés Lépjen a <http://makecode.microbit.org/> webhelyre (vagy egyszerűen kattintson a Kezdőlap ikonra, ha már a MakeCodeEditorban van), és hozzon létre egy új projektet. Kattintson a (Bemenet) kategóriára, majd válassza az (amikor a(z) A gomb lenyomva) blokkot.



2. lépés Kattintson a (Változók) kategóriára, majd válassza a (Változó létrehozása) elemet. Írja be a „kéz” kifejezést a felbukkanó ablakba, majd kattintson az OK gombra.

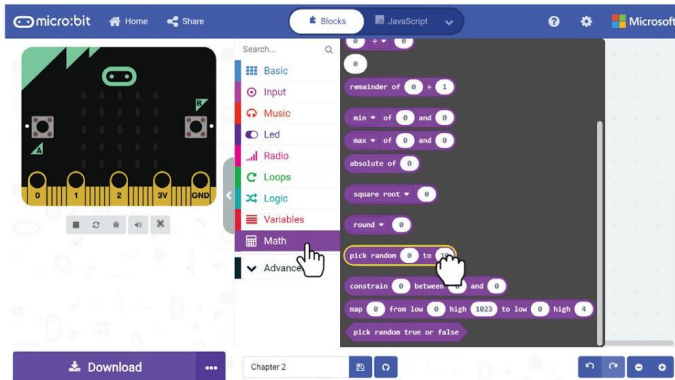


3. lépés Kattintson a (Változók) kategóriára, majd válassza a (kéz legyen) blokkot. Csatlakoztassa a blokkot az (amikor a(z) A gomb lenyomva).

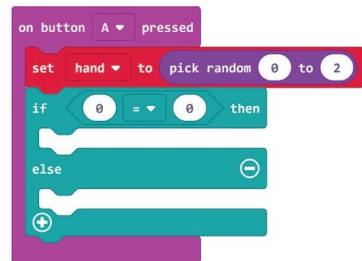
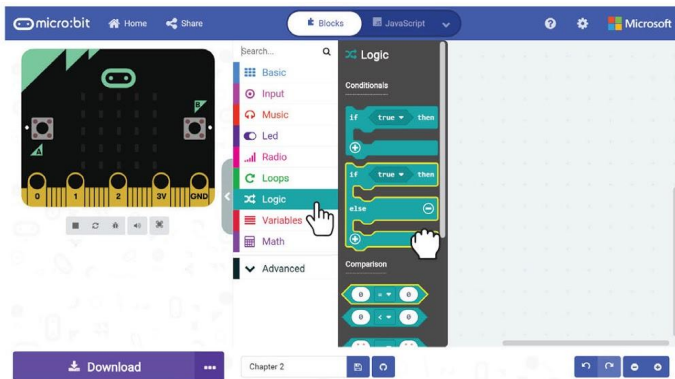




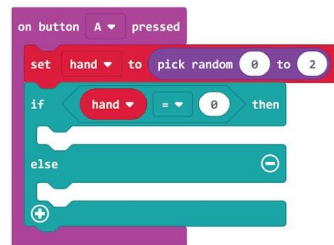
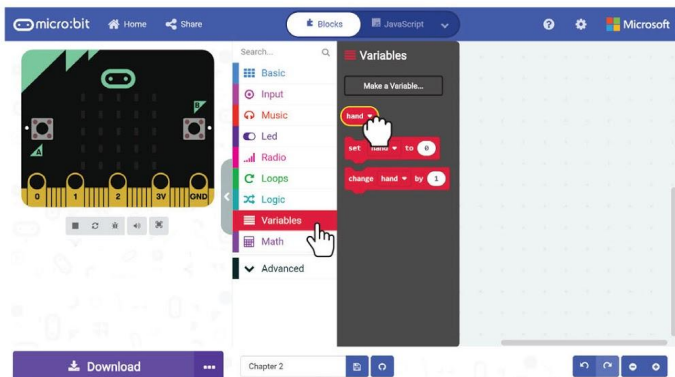
4. lépés Kattintson a (Matek) kategóriára, és válassza a (véletlenszám... és eközött...) blokkot. Változtassa meg a 10-es számot 2-re.



5. lépés Kattintson a (Feltételek) kategóriára, és válassza a (ha-akkor-különben) blokkot és az (... = ...) összehasonlító blokkot. Helyezze az összehasonlító blokkot az 'ha' nyílásba.

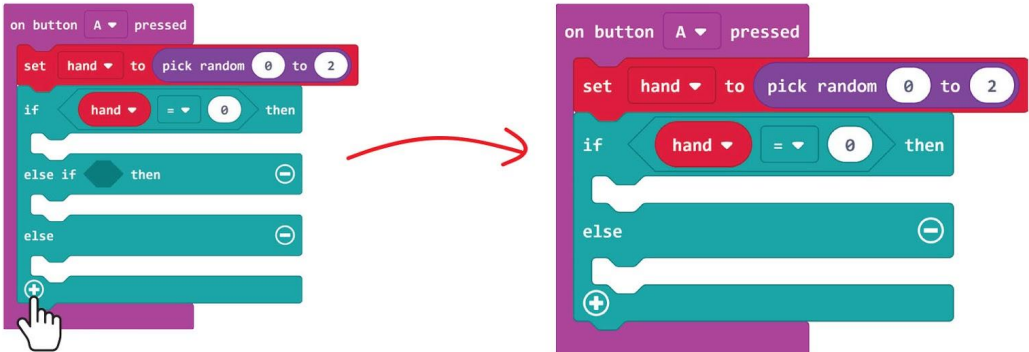


6. lépés Kattintson a (Változók) elemre, majd válassza a (kéz) blokkot. Csatlakoztassa a blokkot az összehasonlító blokkhoz.

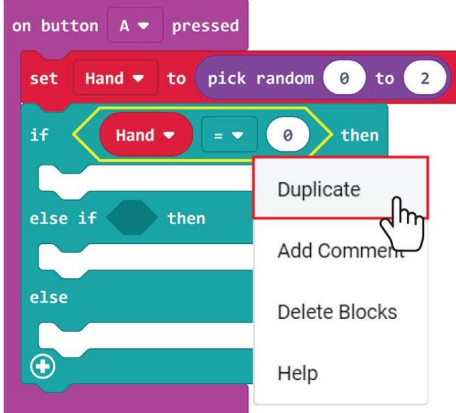


2. Fejezet kő, papír, olló

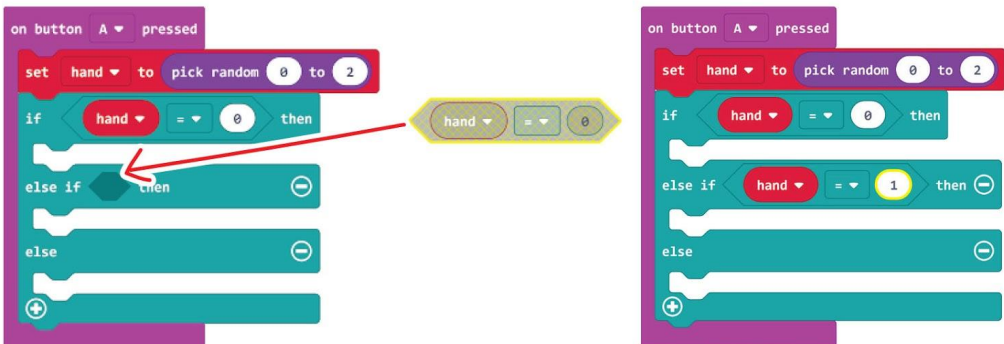
7. lépés: kattintson a + ikonra, hogy hozzáadja a „különben” feltételt a „ha” blokkhoz.



8. lépés Kattintson a jobb egérgombbal az összehasonlító blokkra, majd válassza a „Másolat készítése” lehetőséget.



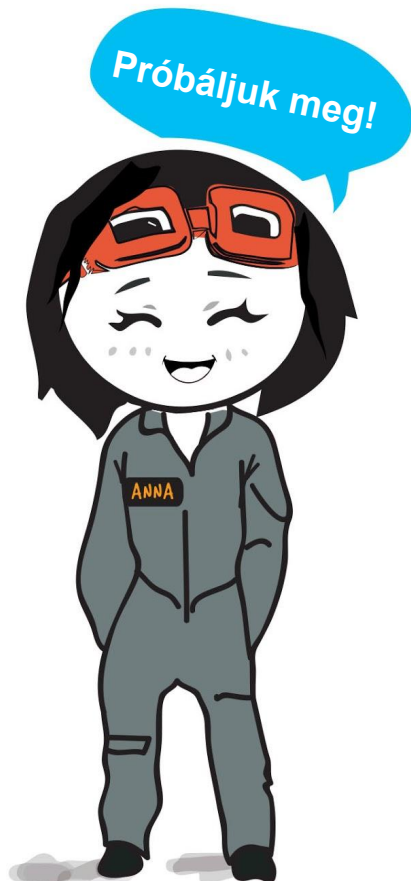
9. lépés Helyezze be a másolatot a 'különben ha_akkor' nyílásba, és változtassa meg a 0-ról 1-re.





10. lépés Add (Basic): (show leds) blokkok az 'if' és 'else' helyekre. Kattintson a (show leds) blokkokban lévő négyzetekre, hogy képeket hozzon létre az alábbi példa szerint:

Próbáljuk meg!



```
on button A pressed
set hand to pick random 0 to 2
if hand = 0 then
  show leds
else if hand = 1 then
  show leds
else
  show leds
```

KŐ

PAPÍR

OLLÓ

Töltsd fel a kódot az EDU:BIT-re és mehet a játék Kő Papír Olló. Bármikor, amikor megnyomja a micro:bit A gombját, vagy a sárga gombot, a Led mátrix véletlenszerűen mutat Követ, Papírt, vagy Ollót.

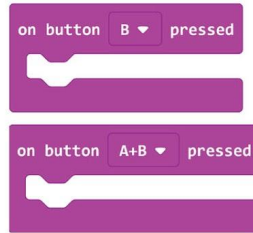
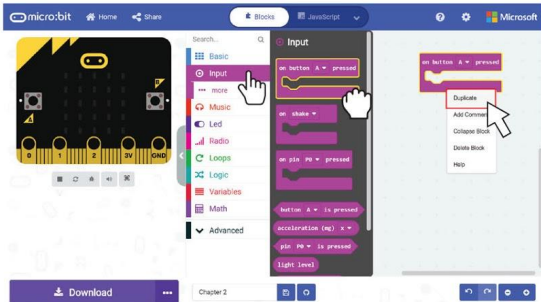
Emlékeztető: Ha meg akarja tartani a projektet, ne felejtse el elmenteni a számítógép egy mappájába a Mentés gombra kattintással.



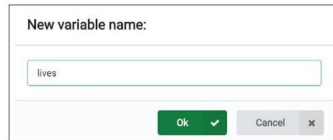
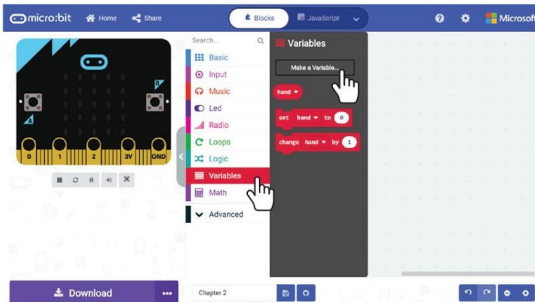


Tegyük fel ezt a játékot egy fokkal magasabbra, azáltal, hogy 3 élete lesz minden játékosnak. Ehhez létre kell hoznia egy új „lives” nevű változót, és hozzá kell adnia a következő kódoló blokkokat.

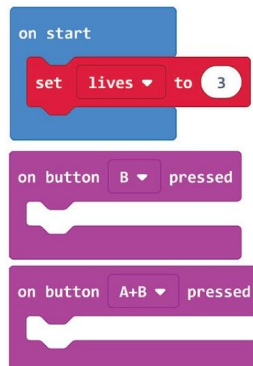
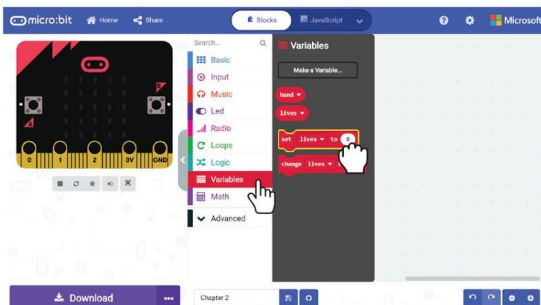
11.lépés Kattintson az (Input) kategóriára, és válassza az (on button_pressed) blokkot. Másolja a blokkot, és változtassa meg a beállítást 'B button'-ra és 'A + B button'-ra.



12. lépés Kattintson a (Variables) kategóriára és válassza a (Make a Variable) elemet. Írja be a „live” elemet a felbukkanó ablakba, majd kattintson az OK gombra.

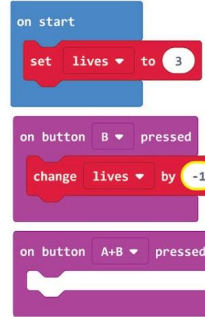
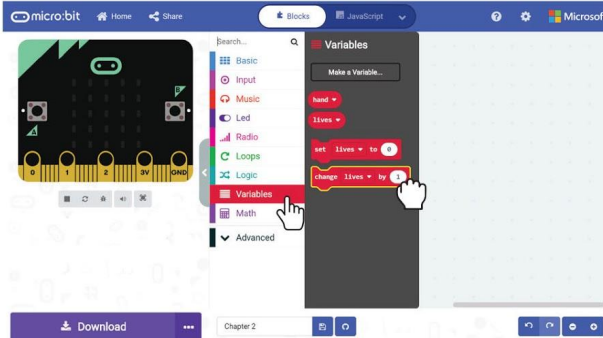


13. lépés Kattintson a (Variables) kategóriára, majd válassza a (set_to_) blokkot. Csatlakoztassa a blokkot a (Basic) :(on start) blokkhoz. Állítsa a változót „lives”, és változtassa meg az értéket 3-ra.

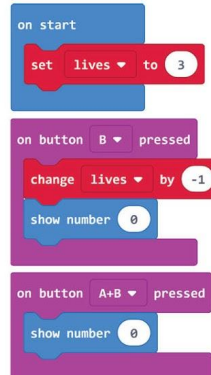
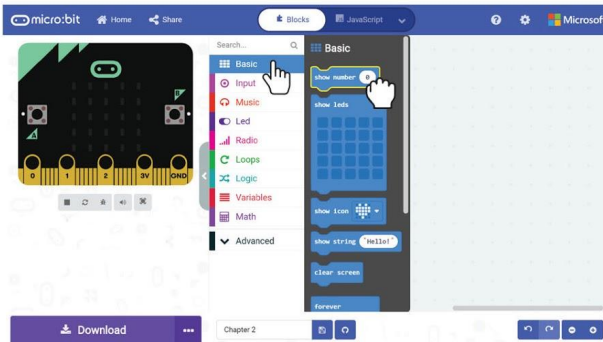




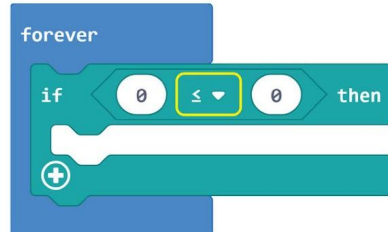
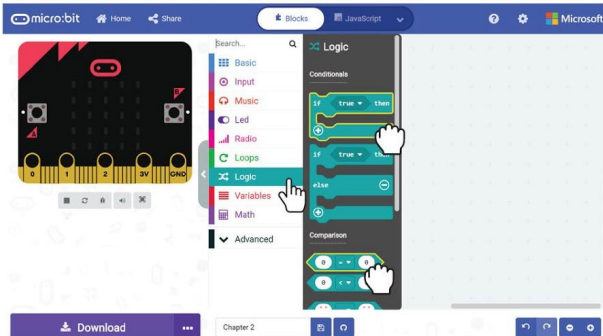
14. lépés Kattintson ismét a **(Variables)** kategóriára, majd válassza a **(change_by_)** blokkot. Csatlakoztassa a blokkot **(on button B pressed)** blokkhoz. Állítsa a változót „lives”, és változtassa meg az értéket -1-re.



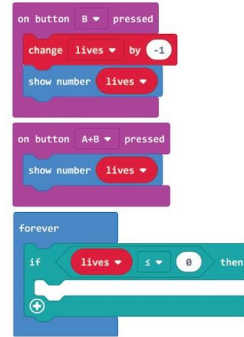
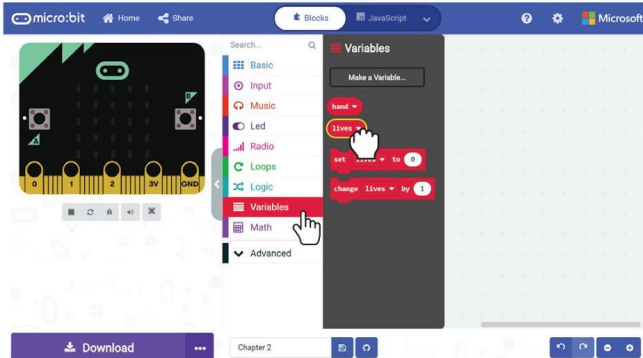
15. lépés Kattintson a **(Basic)** kategóriára, és válassza a **(show number)** blokkot. Másolja be és rögzítse a blokkokat mindkét **(on button A+B pressed)** blokkhoz.



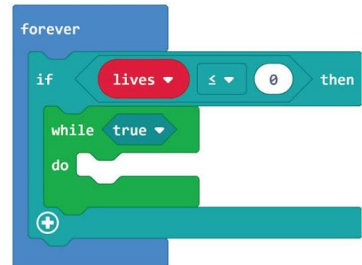
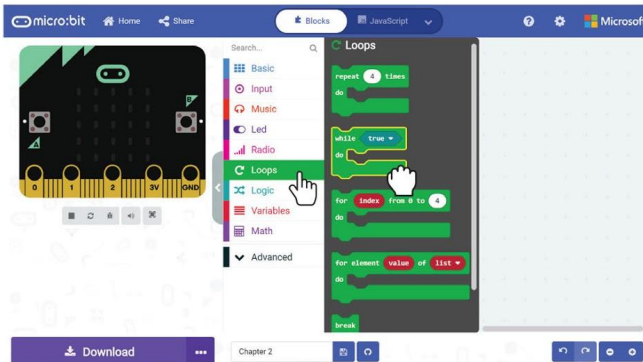
16. lépés Kattintson a **(Logic)** kategóriára, és adjon hozzá **(if-then)** blokkot és **(=)** összehasonlító blokknak a kódjához. Csatlakoztassa őket a **(Basic): (forever)** blokkhoz, és változtassa meg a jelet **'≤'** -re.



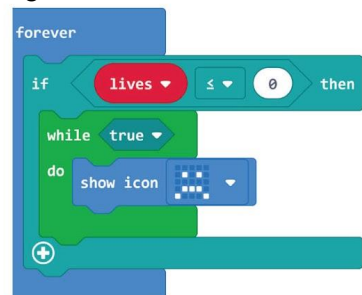
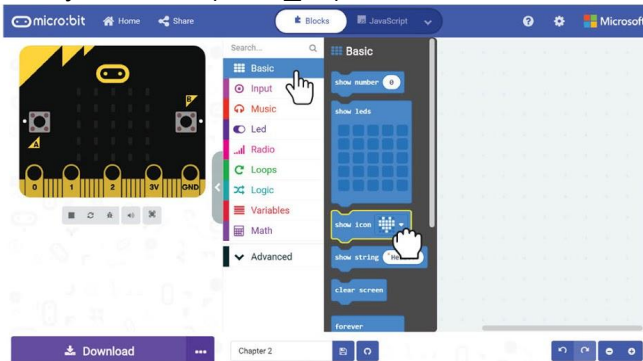
17. lépés Kattintson a **(Variables)** kategóriára, majd válassza az **(Élő)** blokkot. Másolja és illessze be mind a **(szám mutatása)** blokkot, mind az **(_ = _)** összehasonlító blokk bal oldali nyílását.



18. lépés Kattintson a **(Loops)** kategóriára és válassza a **(while_do)** blokkot. Csatlakoztassa az **(if - then)** blokkhoz.



19. lépés Kattintson a **(Basic)** kategóriára és válassza a **(show icon)** blokkot. Helyezze rá a **(while_do)** blokkra, és változtassa meg az ikont „szomorú arcra”.





20. lépés Itt a teljes kód. Töltsd rá az EDU:BIT-re, és érezd jól magad a barátáiddal, hogy ki a király/királynő a Kő Papír Olló játékban.

```
on button A pressed
  set hand to pick random 0 to 2
  if hand = 0 then
    show leds
  else if hand = 1 then
    show leds
  else
    show leds
  forever
    if lives <= 0 then
      while true
        do show icon
```

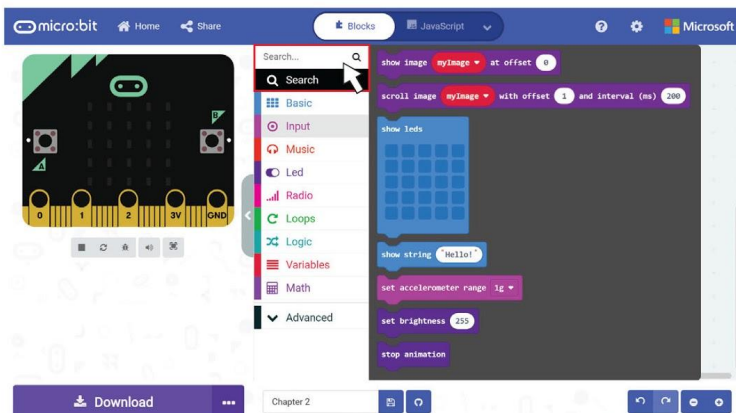
```
on start
  set lives to 3

on button B pressed
  change lives by -1
  show number lives

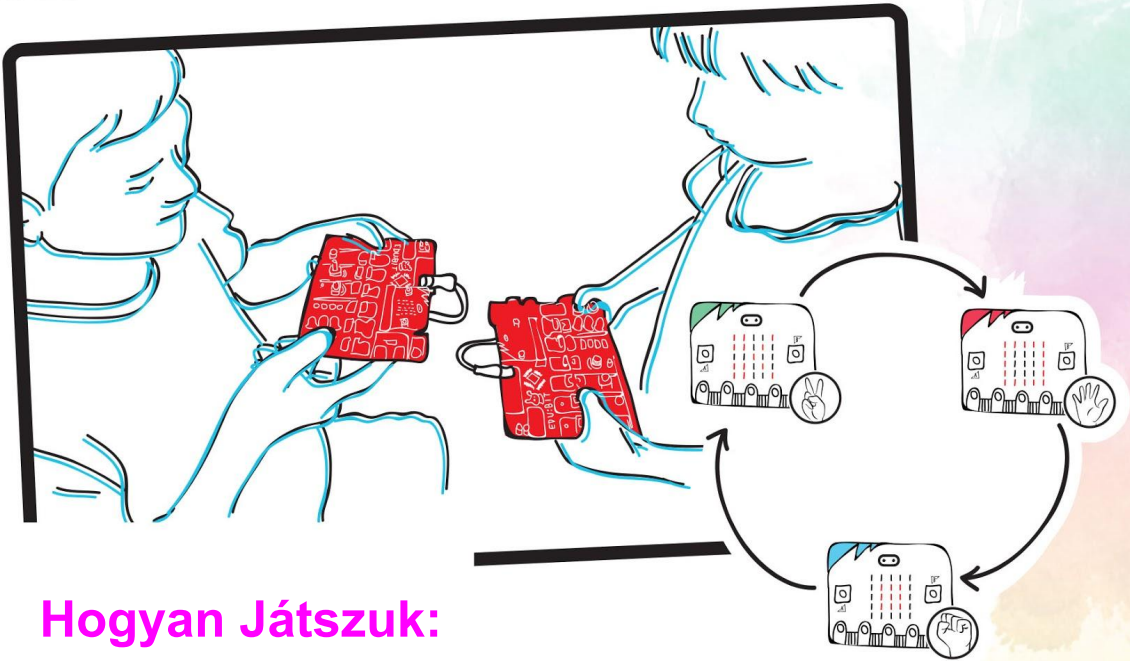
on button A+B pressed
  show number lives
```

Itt egy tipp az Ön számára...

Minden kódoló blokk színkóddal van ellátva. Keress meg a szükséges blokkokat a kategóriaфиókban, azonos színűek. Alternatív megoldásként beírhatja a Kulcsszavak elemét a keresőmezőbe.



Játszunk!



Hogyan Játsszuk:

- Álljon szemben az ellenféllel. Amikor mindkét játékos készen áll, nyomja meg a sárga gombot (A gomb), hogy véletlenszerűen megjelenjen a kő, papír, olló.
- Hasonlítsa össze és döntse el, hogy ki nyer.
- Nyomja meg egyszerre a sárga és a kék gombot a fennmaradó életek számának ellenőrzéséhez és megjelenítéséhez.
- Ha egy játékos háromszor veszít, akkor Game Over lesz, és az EDU:BIT szomorú arcot fog mutatni.

Megjegyzés!

- Egy másik forduló lejátszásához vissza kell állítania a táblát az újra indításhoz.
- Ha nincs barátja, akivel játszhat, akkor mindig játszhat a szimulátor ellen a MakeCode Editorban.

Értelmezzük a program működését

A számítógépes programozás során változókat használunk olyan információk vagy értékek tárolására, amelyek megváltoztathatók futási időben (amikor a program fut). Gondolhat egy változóra, mint egy címkézett borítékra, amely egy darab papírt tartalmaz, amelyre rá vannak írva információk. A papírdarabot ki lehet venni, és helyettesítheti egy újabb papírral, új információkkal.

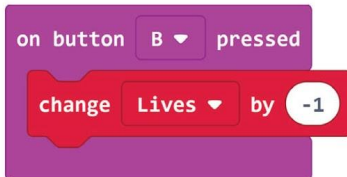
Kódunkban korábban létrehoztunk egy "lives" nevű változót, és hozzárendeltünk 3 kezdeti értékét.



A darab papírra írt 3. szám az információ.

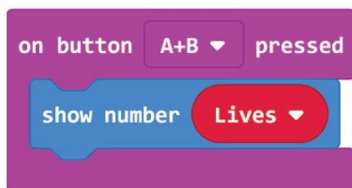
A rá írt „élet” boríték változóként ismert.

Ezután minden B gomb megnyomásakor az értéket -1-el megváltoztatja

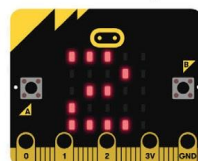


A B gomb megnyomásakor a "3" betűvel ellátott darabot eltávolítják a borítékból, és helyettesítik egy másik "2" betűvel ellátott papírral.

Az A + B gomb egyidejű megnyomásával a Led mátrix megjeleníti a változó élettartamának aktuális értékét.



Amikor megnyomja az A + B gombot, vegye ki a papírlapot a borítékból, és "olvassa" a rajta található információkat.

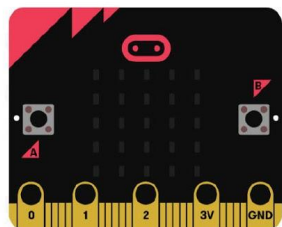
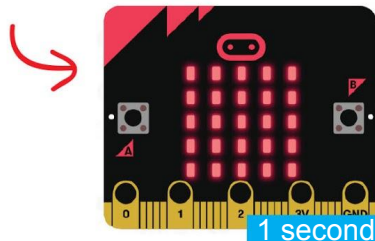
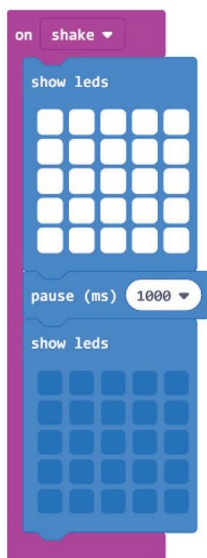
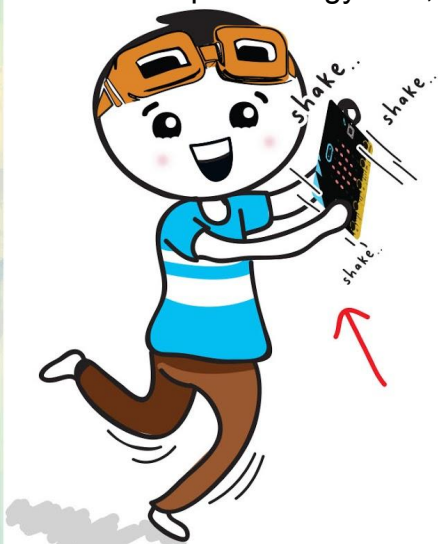




Fedezzen fel további blokkokat

A [on button_pressed] blokk mellett az [input] csoportból más blokkokat is használhat eseményalapú programozáshoz. A felhasználó által végrehajtott tevékenységeket, például egy gombnyomást vagy a tábla rázását, eseménynek nevezzük.

A következő kód például azt eredményezi, hogy a LED-mátrix 1 másodpercre felgyullad, amikor a tábla rázkódik. Próbáld ki



Ha rákattint a blokk [shake] gombjára, megjelenik egy felbukkanó menü, amelyben megjelenik a kiválasztott egyéb eseményindítók száma. Próbálkozzon az EDU: BIT programozásával, hogy ezeknek a feltételeknek mindegyikére más ikon jelenjen meg. Jó szórakozást felfedező!



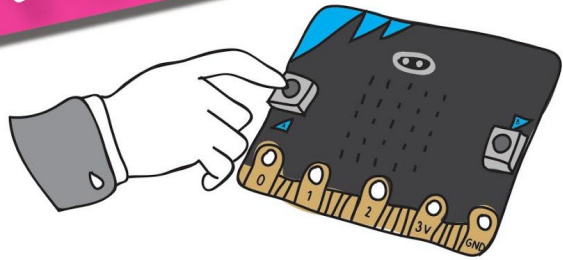
EDU:BIT érzékeli a mozgó eseményeket és ismeri a saját tájolását, mivel van egy beépített mozgásérzékelő a micro:bit-ben.



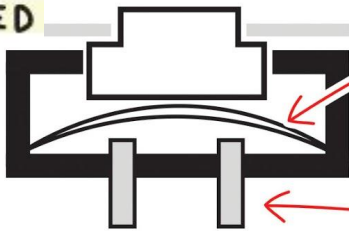
Érdekesség!



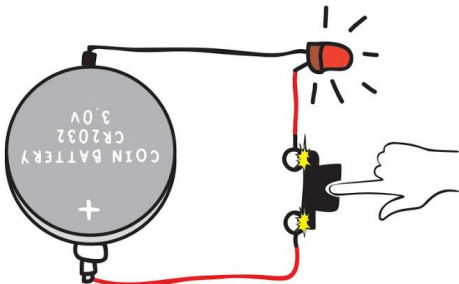
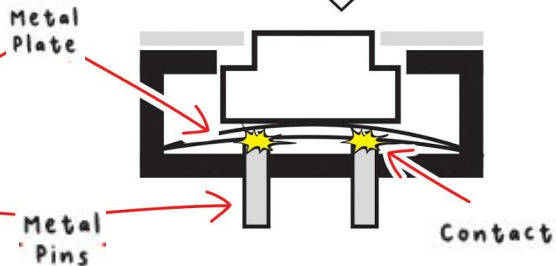
A nyomógomb egy bemeneti eszköz vagy kapcsoló, amelyben csak két lehetséges állapot van megnyomva vagy nem megnyomva.



NOT
PRESSED



PRESSED



A nyomógomb megnyomása után befejezi az elektromos áramkört, és a LED világít! Találd ki, mi történik, amikor elengedi a nyomógombot?

HELP

PANIC
BUTTON



PRESS BUTTON
WHEN YOUR SAFETY
IS THREATENED



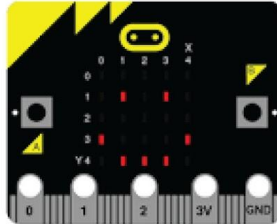
A BE, KI funkcióhoz általában a fekete, a szürke, a zöld és a fehér gombokat, a pánikgombokhoz vagy a gépek veszkikapcsolásához a Pirosat használják.

Learn
more!



youtu.be/t_Qujjd_38o

Alkalmazás Kihívás



A EDU:BIT program számlálóként szolgál a hallgatói részvétel nyilvántartására. A lányoknak meg kell nyomniuk az A gombot, amikor belépnek az osztályba; és a fiúknak meg kell nyomniuk a B gombot

Indításkor	Mutasson egy mosolyt. Állítsa be a Lányok = 0 és a Fiú = 0 változókat
A gombnál (Sárga gomb)	Változtassa (növelje) meg a Lány változót 1-gyel
B gombnál (Kék gomb)	Változtassa (növelje) meg a Fiú változót 1-gyel
A+B gombnál	Görgesse a következő információt a LED-kijelzőn: Összesen= (Lány+Fiú), Lány(Lány), Fiú(Fiú)

Zenéljünk
Music Bit (Piezo Buzzer + Audio Jack)



Scan Me!

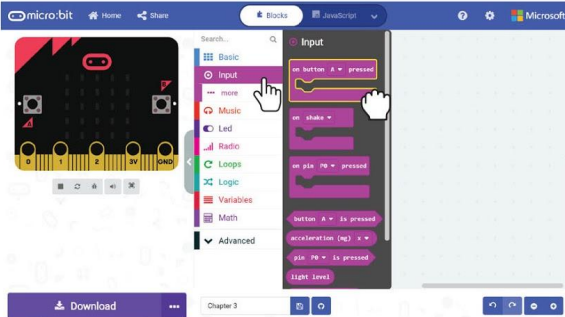


link.cytron.io/edubit-chapter-3

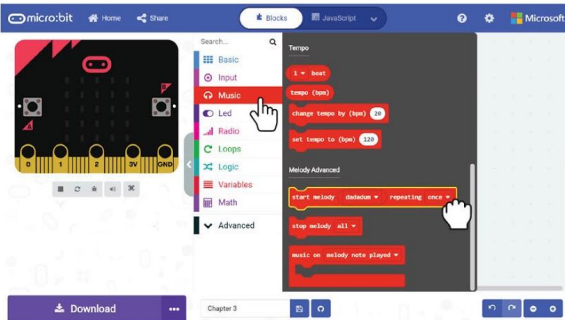
3. fejezet : Készítsünk zenét

Kódoljunk!

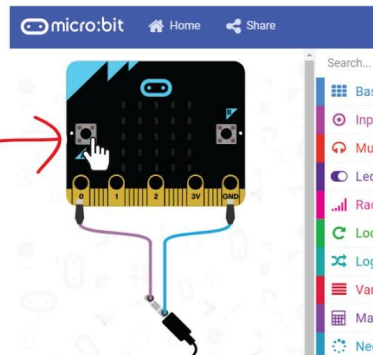
1.lépés Hozzon létre egy új projektet a MakeCode Editorban. Kattintson az [Input] kategóriára, majd válassza a [on button_pressed] blokkot.



2.lépés Kattintson a [Music] kategóriára, majd válassza a [start melody_repeating_] blokkot.



3. lépés Kattintson a (dadadum) elemre, és válassza a „birthday” dallamot a legördülő listából.



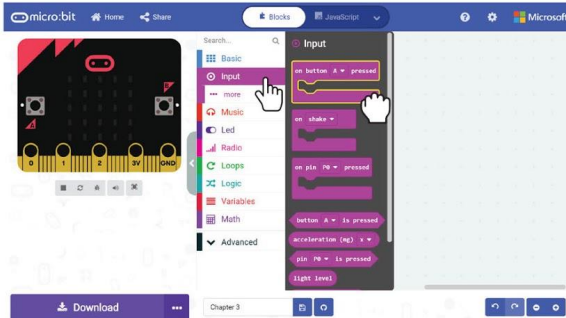
Kattintson a képernyő-szimulátor A gombjára. Hall egy ismerős dallamot? Jó szórakozást a többi dallam kipróbálásában is *Ellenőrizze, hogy a számítógép hangszórói be vannak-e kapcsolva.



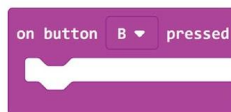
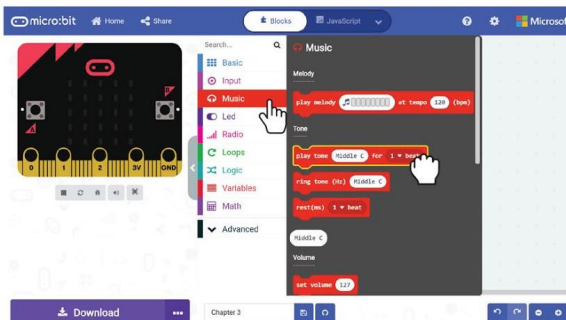
Az előre beállított dallamok listáján kívül az EDU: BIT programozható bármilyen kívánt dal lejátszására. Próbáljuk meg lejátszani ezt a fülbemászó dallamot



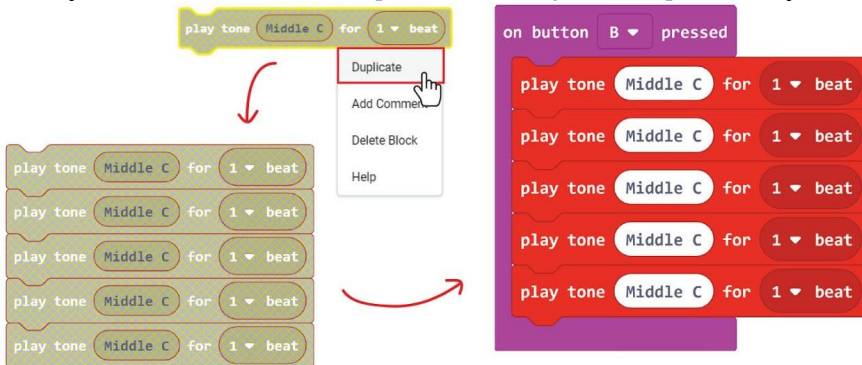
4. lépés Kattintson az [Input] kategóriára, majd válassza a [on button_pressed] blokkot. Válassza a "B" gombot.



5. lépés Kattintson a [Music], majd válassza a (play tone_for_beat) blokkot.

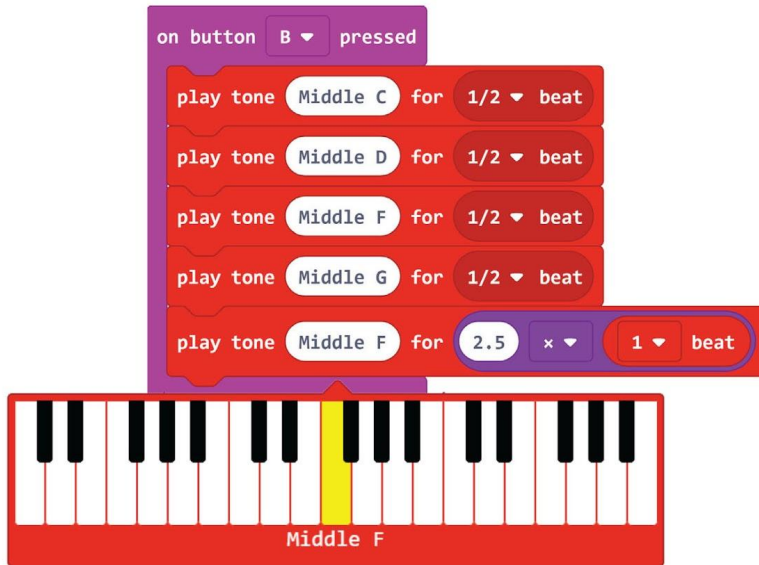


6. lépés A munkaterületen kattintson a jobb gombbal a [play tone_for_beat] blokkra, majd kattintson a 'Duplicate' gombra. Addig ismételje meg, amíg összesen öt [play tone_for_beat] blokkot nem kapsz. Helyezze a blokkokat az [on button A pressed] blokk nyílásába.



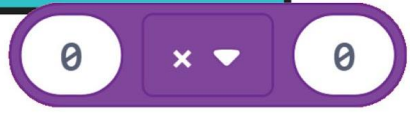
3. fejezet : Készítsünk zenét

7. lépés Válassza ki a „tone” és a „beat” [play tone_for_beat] blokkokat az alábbi mintakód szerint:



Megjegyzés!

A lila blokk a [Math] kategóriájú fiókból származik.



Kattintson a képernyő-szimulátor B gombjára. Meg tudja tippelni a dalt?



A programozás során tanácsos rendszeresen ellenőrizni a kódját, hogy megbizonyosodjon arról, hogy jó úton halad-e. Erre a célra használhatja a szimulátort.

8. lépés: Folytassa a dal többi részének kódolását további blokkok hozzáadásával [play tone_for_beat] és ennek megfelelően a „tone” és a „beat” megváltoztatásával. A hangok és a csengőhangok következő oldalára hivatkozhat.



I Will Follow You

on button B ▾ pressed

I will follow you,

Follow you wherever you may go,

There is n't an ocean too deep

play tone Middle C for 1/2 ▾ beat
 play tone Middle D for 1/2 ▾ beat
 play tone Middle F for 1/2 ▾ beat
 play tone Middle G for 1/2 ▾ beat
 play tone Middle F for 2.5 x ▾ 1 ▾ beat
 rest(ms) 1/2 ▾ beat
 play tone Middle C for 1/2 ▾ beat
 play tone Middle D for 1/2 ▾ beat
 play tone Middle F for 1/2 ▾ beat
 play tone Middle D for 1/2 ▾ beat
 play tone Middle F for 1/2 ▾ beat
 play tone Middle A for 1/2 ▾ beat
 play tone High C for 1.5 x ▾ 1 ▾ beat
 play tone Middle A for 1/2 ▾ beat
 play tone High C for 2 ▾ beat
 rest(ms) 1/2 ▾ beat
 play tone High C for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone Middle A for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone High C for 2 ▾ beat
 rest(ms) 1/2 ▾ beat

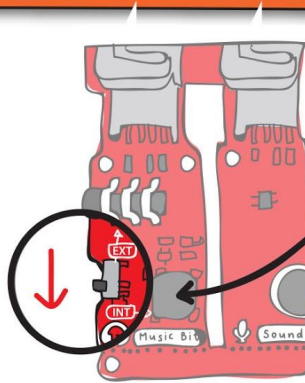
A mountain so high it can keep, Keep me a-way ...

play tone High C for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone High C for 1/2 ▾ beat
 play tone Middle B for 1/2 ▾ beat
 play tone High C for 2 ▾ beat
 rest(ms) 1/2 ▾ beat
 play tone High C for 1/2 ▾ beat
 play tone Middle A for 1 ▾ beat
 play tone Middle G for 1 ▾ beat
 play tone Middle A for 1 ▾ beat
 play tone Middle G for 1/2 ▾ beat
 play tone Middle F for 2.5 x ▾ 1 ▾ beat
 rest(ms) 1 ▾ beat



9. lépés A kitöltött kód villogása az EDU: BIT-re.

EDU:BIT lejátsza az "I will follow you" dalt, amikor megnyomja az EDU: BIT kék gombját (B gomb). Ne felejtse el bekapcsolni az EDU:BIT-et, és kapcsolja be a piezo-hangszórót a kapcsoló INT (belső) állásba csúsztatásával.

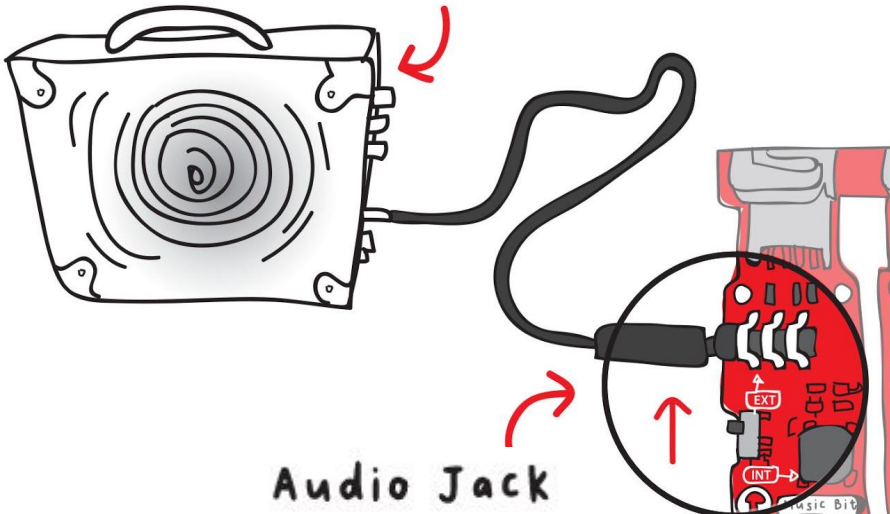


Piezo Buzzer



Alternatív megoldásként külső hangszórókat vagy fejhallgatókat is csatlakoztathat az EDU:BIT audio aljzathoz. A kapcsolót EXT (külső) állásba kell csúsztatnia.

Speakers

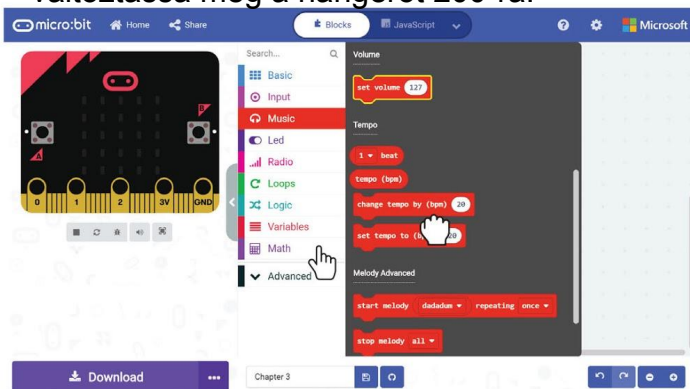


Audio Jack



Túl gyenge a zene? Vagy túl hangos? Fel vehet egy (set volume_) blokkot a hangerő 0 és 255(max hang) közötti tartományának megváltoztatásához

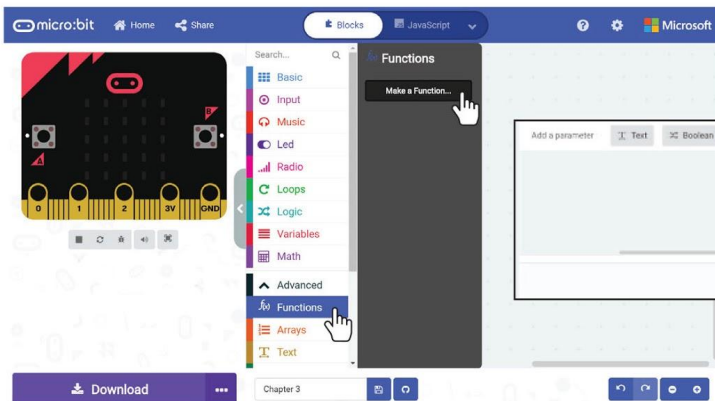
10. lépés Kattintson a (Music) kategóriára, és válassza a [set volume] blokkot. Csatlakoztassa a blokkot (on start) blokkra, és változtassa meg a hangerőt 200-ra.



Megjegyzés!

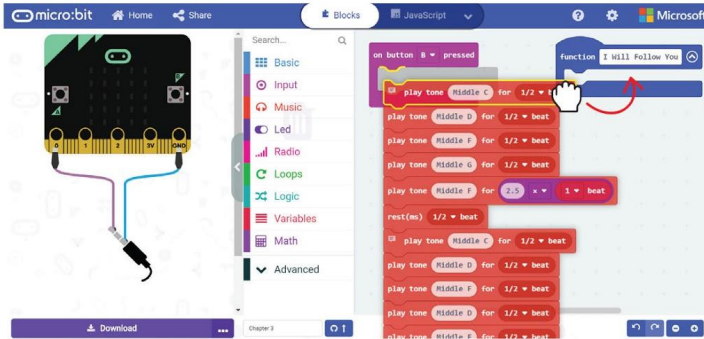
Kidolgozhatunk olyan kódblokkokat, amelyek egy adott feladatot végrehajtanak, például a kódját, hogy lejátsszuk a "I Will Follow You" dalt. A programozás során egy funkció rutinra vagy eljáráskészletre utal. Miután egy funkció meg lett határozva, több helyen használható a programban anélkül, hogy újra és újra meg kellene írni ugyanazon kódblokkokat.

11.lépés Kattintson a [advanced] kategóriára, majd válassza a [Functions] kategóriát. Kattintson a [Make a Function] elemre, és nevezze át a doSomething elemet „I Will Follow You” névre a felbukkanó ablakban. Ezután kattintson a 'Done' gombra.

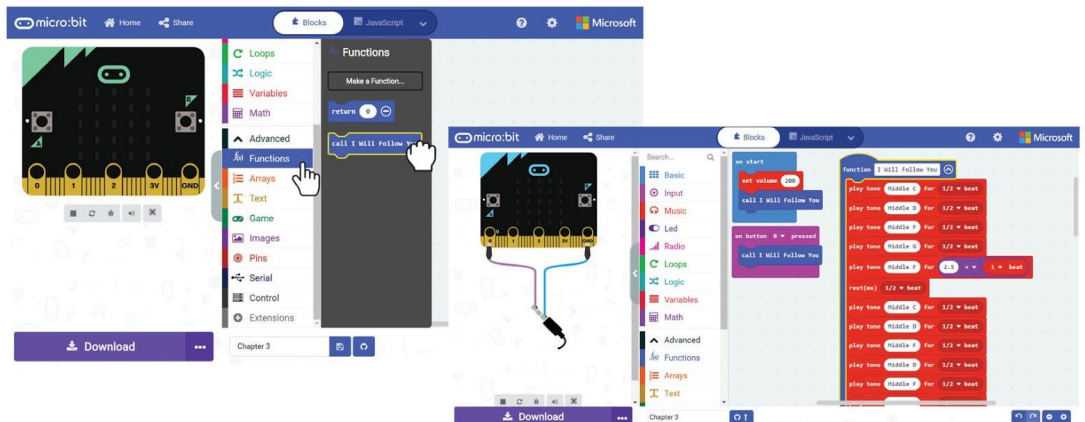


3. fejezet : Készítsünk zenét

12.Lépés (function I Will Follow You) blokk jelenik meg a Szerkesztőben. Kattintson a [on Button B pressed] blokkjának legfelső blokkjára, tartsa lenyomva és húzza az összes blokkot a (function I Will Follow You) nyílásba.



13. lépés Kattintson a (Functions) kategóriára, majd válassza a (call I Will Follow You) blokkot. Másolja a blokkot. Csatlakoztassa a (call I Will Follow You) blokkokat az (on start) és a (on button B pressed) blokkhoz. Íme a példakód:



14. lépés: Töltsd át a teljes kódot az EDU: BIT-re. Élvezze a zenét~

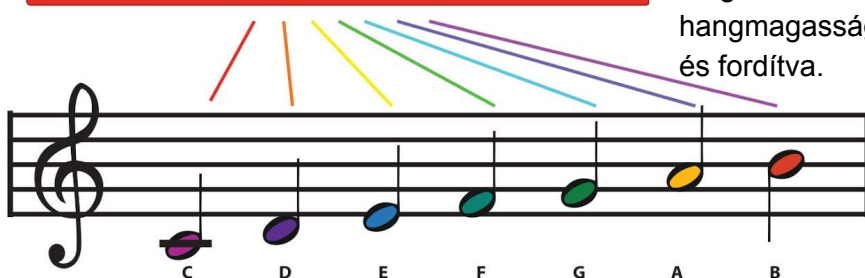
Értelmezzük a program működését



Az EDU: BIT programozható más dalok lejátszására, ha tudod, hogyan kell kottát olvasni. Ez egy egyszerű útmutató, amely segít egy zenei partitúra "dekódolásához".



A hangjegy elhelyezkedése a kottában (azaz az öt vízszintes vonal) megmutatja, hogy melyik hangot kell lejátszani. Minél magasabb a hang, annál nagyobb a hangmagasság vagy a frekvencia, és fordítva.



Sign	Rest	Relative Length	Duration
		Whole Note	4 beats
		Half Note	2 beats
		Quarter Note	1 beat
		Eighth Note	1/2 beat
		Sixteenth Note	1/4 beat

Különböző zenei jelöléseket használnak, hogy megmondják nekünk, hogy a hang lejátszása milyen hosszú.

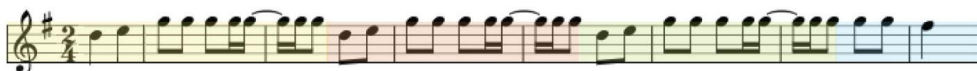
Értelmezzük a program működését

A megadott forrást felhasználva "dekódolható" a következő dallam?



Baby Shark

$\text{♩} = 115$



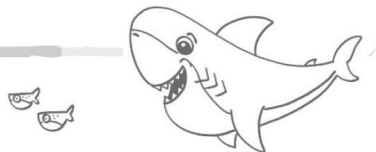
Line 1	Ba	-by	Shark	doo	doo,	doo	doo	doo	doo
Note	High D	High E	High G		High G	High G	High G		High G
Beat	1		1/2	1/2	1/2		1/2	1/4	1/2

Line 2	Ba	-by	Shark	doo	doo,	doo	doo	doo	doo
Note	High D		High G	High G	High G		High G	High G	High G
Beat	1/2	1/2	1/2		1/2	1/4	1/2		1/2

Repeat the same for Line 3

Line 4	Ba	-by	Shark
Note		High G	High F#
Beat	1/2	1/2	

Program EDU:BIT to play Baby Shark tune when the yellow button (Button A) and blue button (Button B) are pressed at the same time.



Megjegyzés!

A [set volume_] blokk segítségével állítsa be a dallam hangerejét



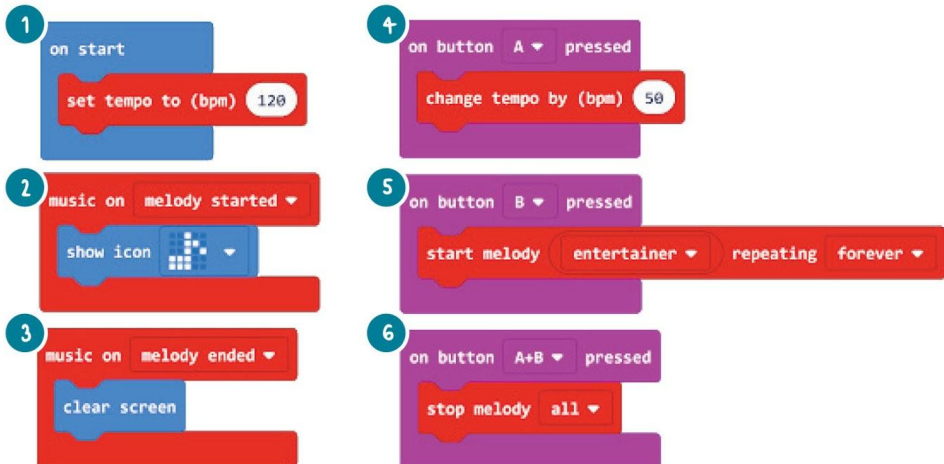
Fedezzen fel további blokkokat!

1 Beállíthatja a „tempo”-t (azaz a dal ütemét) a **[set tempo to (bpm)]** blokk segítségével. Minél magasabb a percenkénti sebesség (ütés percenként), annál gyorsabb vagy élénkebb a dallam. A tempó megváltoztatásához használja a **[set tempo by (bpm)]** blokkot.

2 A **[stop melody_]** blokk segítségével állítsa le a jelenleg lejátszott dallamot.

3 Használhatja a **[music on_]** és annak feltételeinek kiválasztását is, például a dallam elindulását és a dallam végét, mint eseményindítót a kódjában.

Íme egy egyszerű kód:

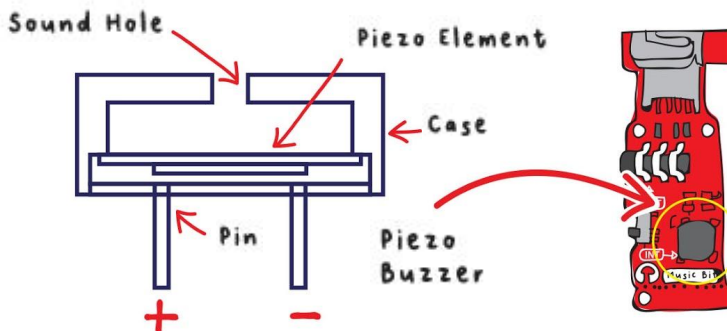


- 1 Ebben a programban a kezdeti tempót 120 bpm-re állítja.
- 2 A dallam elindulásakor a LED mátrix megjeleníti a hangjegy ikont.
- 3 A dallam befejeztével a mátrix kijelzőjén az összes LED kialszik.
- 4 Az „A” gomb minden egyes megnyomására a tempót 50 bpm-rel növelik
- 5 Az „entertainer” dallamot mindig a „B” gomb megnyomásakor kell lejátszani.
- 6 A dallam leáll, ha az „A + B” gombot egyszerre nyomja meg.

Érdekesség!

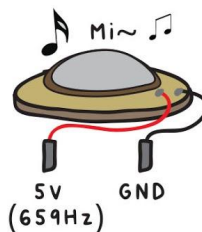
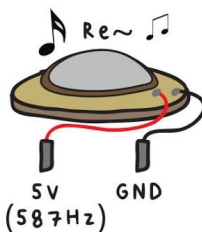
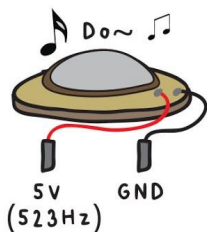


A **piezo-hangjelzőt** általában a hang előállításához használják, amikor egy darab piezo-elem rezeg, akkor az elektromos jel áthalad rajta.

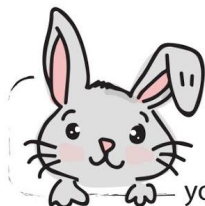


Az elektromos jel frekvenciájának megváltoztatásával a rezgések sebessége megváltozik; és így a piezo más hangjelzéssel állít elő hangot.

Piezo elem

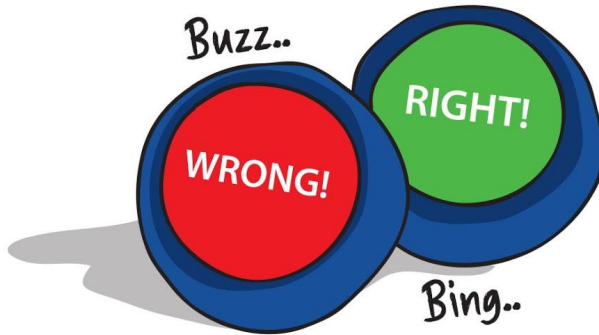


Az emberi fül 20 Hz - 20 000 Hz frekvenciatartományban hallható. A 20 Hz alatti hangot infrahangnak nevezzük, és minden, ami 20 000 Hz felett van, ultrahangnak tekintjük.



youtu.be/cxfPNc4Wefo

Alkalmazás kihívás



EDU:BIT Game Show Buzzer működik a helyes / helytelen válaszok jelzésére.

On start	Jelenítsen meg egy mosolygó arcot
„A” gomb/ Sárga gomb	Jelenítsen meg egy ✓ ikont, és játssza le egyszer a „power up” dallamot
„B” gomb/ Kék gomb	Jelenítsen meg egy ✗ ikont, és játssza le egyszer a „wawawaaa” dallamot
„A+B” gomb	Törölje a képernyőt.

Nyersz, Veszítesz vagy Rajzolsz
Traffic Light Bit



Pulyka!



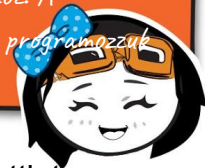
átnázheted



link.cytron.io/edubit-chapter-4



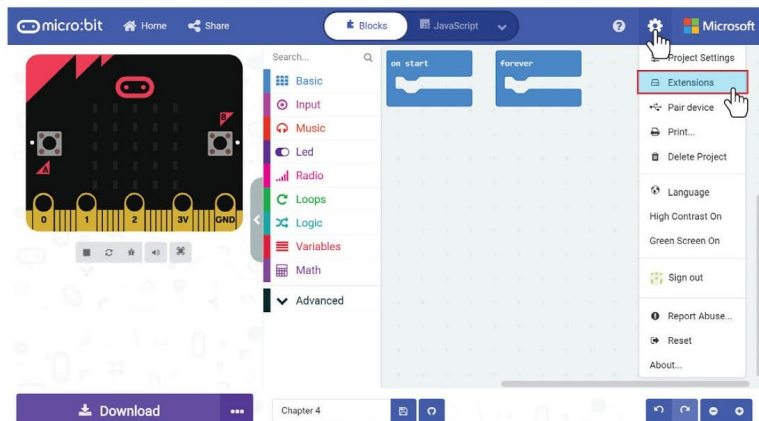
Észrevetél egy sor piros, sárga és zöld LED-et az EDU: BIT-en? Ez a Traffic Light Bit. A programozáshoz hozzá kell adnia az EDU: BIT kiterjesztést a MakeCode Editor-hoz. A kiterjesztések olyan blokkok, amelyeket hozzáadunk a szerkesztőhöz, hogy könnyen programozzuk a micro: bit kiegészítőket, például az EDU: BIT-et.



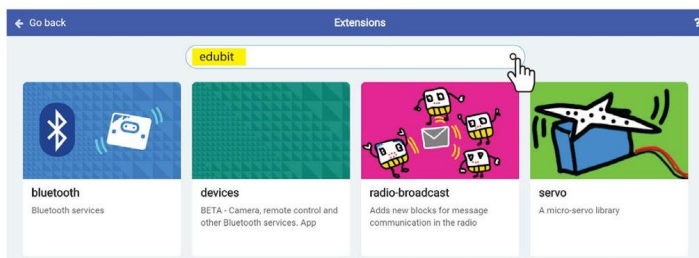
Kódojunk!

1.lépés Hozzon létre egy új projektet a MakeCode Editorban. Kattintson a fogaskerék ikonra, és válassza az „Extensions” lehetőséget.

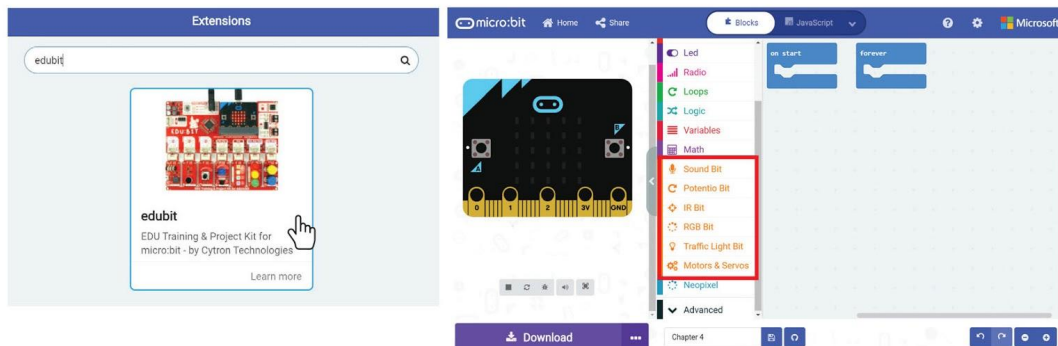
* Internet-kapcsolat szükséges a bővítmények hozzáadásához.



2.lépés Írja be az „edubit” kifejezést a keresőmezőbe, és katt. az Enter gombra.

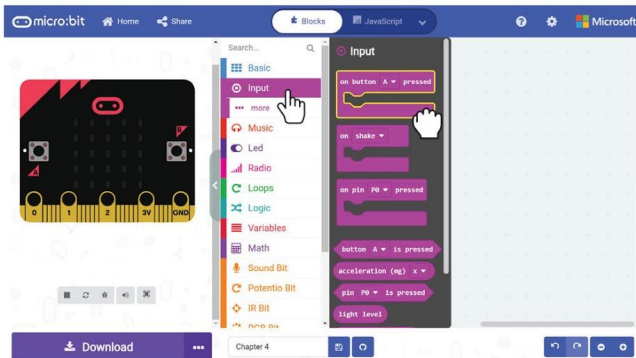


3.lépés Kattintson az 'edubit' kiterjesztésre. Várja meg, amíg betöltődik, és észreveszi a következő új kategóriákat a MakeCode Editorban.

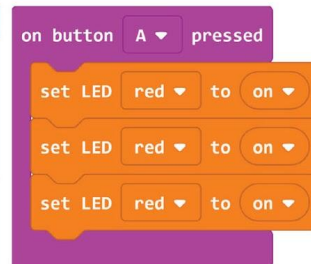
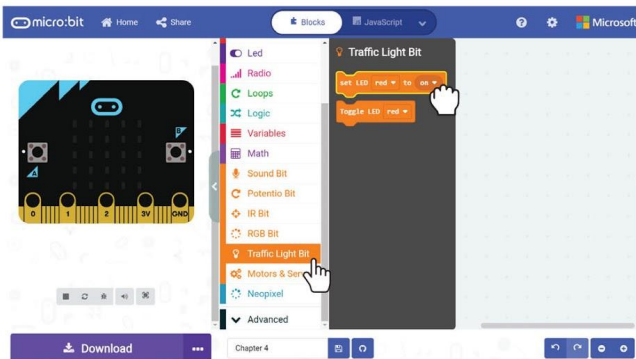


4. fejezet : Traffic Light Bit

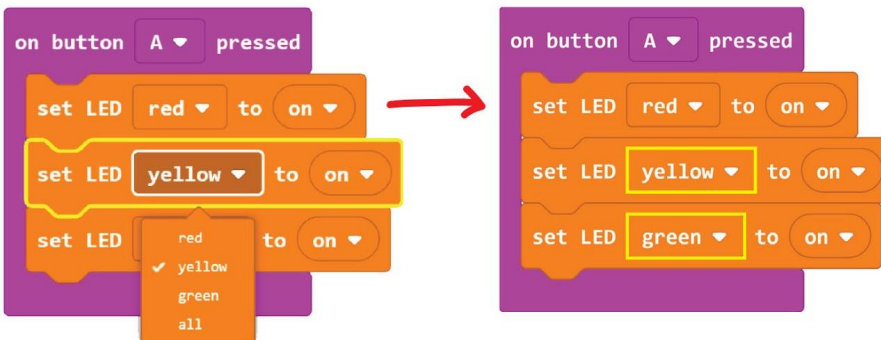
4.lépés Kattintson az [Input] kategóriára, majd válassza a [on button_pressed] blokkot.



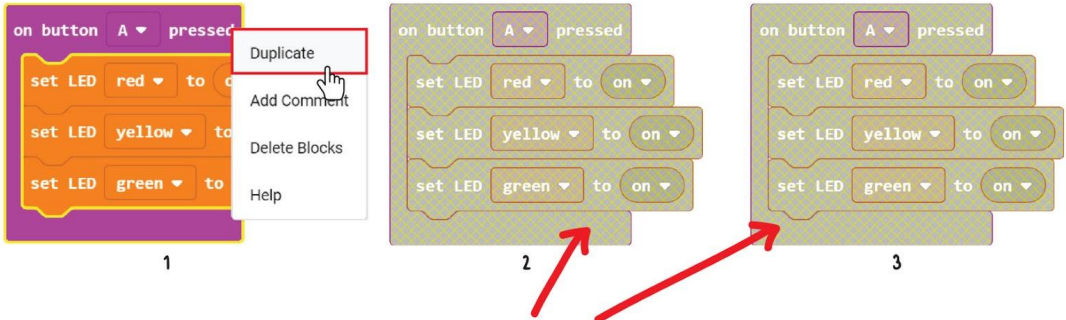
5.lépés Kattintson a [Traffic Light Bit] kategóriára, majd válassza a [set LED_to_] blokkot. A Munkaterületen kattintson a jobb gombbal a [set LED_to_] blokkra, majd kattintson a 'Duplicate' elemre. Addig ismételje meg, amíg három [set LED_to_] blokk nem lesz. Csatlakoztassa a blokkokat az [on button A pressed] nyílásba.



6.lépés Kattintson a színválasztásra és változtassa meg a második és a harmadik blokkot 'sárga' és 'zöld' értékre.



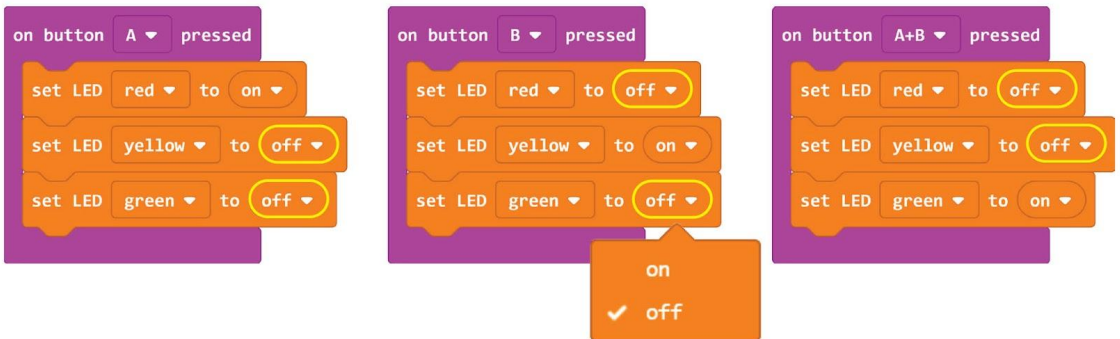
7.lépés Kattintson a jobb gombbal az (on button_pressed) blokkra, majd válassza a "Duplicate" lehetőséget. Ismételje meg, hogy ugyanazon blokkok három halmazát kapja meg.



* Ezek a blokkok le vannak tiltva és nem fognak futni, mert több a (on button A pressed) blokk.

8.lépés Változtassa meg az "A" -t a második és a harmadik (on button pressed) blokkokban "B" -re és "A + B" -re.

9.lépés Kapcsolja be és ki a LED-ek állapotát, az alábbiak szerint.



10.lépés Töltse fel a kódot az EDU:BIT-re, és figyelje meg, hogy mi történik, ha megnyomja az A, a B gombot, majd a két A + B gombot.



Maradj kint!

Váraj.

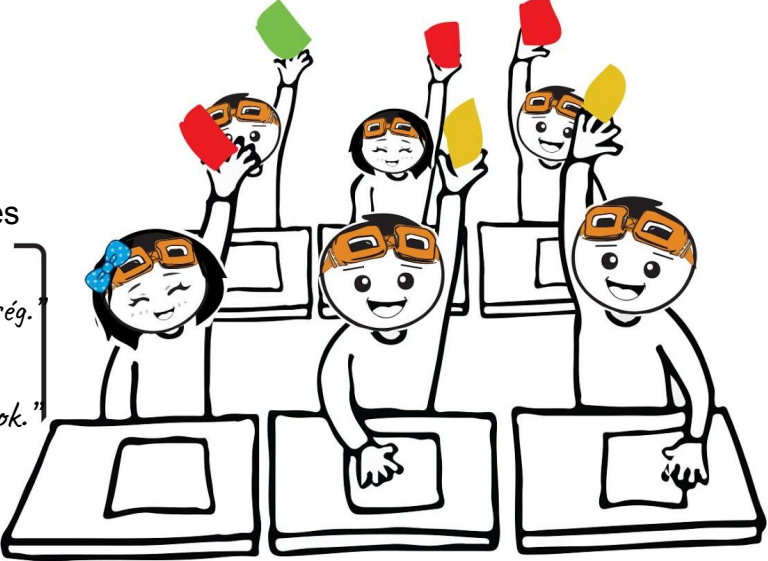
Gyere be.

Ju hú ... most már használhatja az EDU: BIT-et személyes jelzésekre. Akár másra is felhasználhatja?



Jelzőlámpa visszajelzés

-  „Elakadtam! Segítség.”
-  „Még dolgozok rajta/ próbálkozok.”
-  „Mehet!”



A LED, vagy a fénykibocsátó dióda egy példa a digitális kimeneti eszközre. Csak két lehetséges állapota van - be vagy ki? Bekapcsoltnál: 1 (egy), Kikapcsoltnál: 0 (nulla)



Azt is programozhatja, hogy az **EDU: BIT** időmérőként működjön. Itt van a minta kódja.



Az időzítő az EDU: BIT megrázásakor aktiválódik

A hangjelzés jelzi, hogy az időzítő elindult

A zöld LED felkapcsol

A sárga LED felkapcsol.

A piros LED felkapcsol.

Ha az idő lejárt, játssza le a "wawawaaa" dallamot.

Ismételje meg 10-szer a pirosra váltást.

```
on shake
  play tone Middle C for 1 beat
  set LED red to off
  set LED yellow to off
  set LED green to on
  pause (ms) 2000
  set LED red to off
  set LED yellow to on
  set LED green to off
  pause (ms) 2000
  set LED red to on
  set LED yellow to off
  set LED green to off
  pause (ms) 2000
  start melody wawawaaa repeating once
  repeat 10 times
    do
      Toggle LED red
      pause (ms) 500
```

Ebben a mintakódban az egyes LED-ek 2000 ms (2 másodperc) alatt világítanak.

Ha azt szeretné, hogy minden LED 1 percig világítson, milyen értéket kell itt megadnia? Íme egy tipp az Ön számára:

1 perc = 60 másodperc

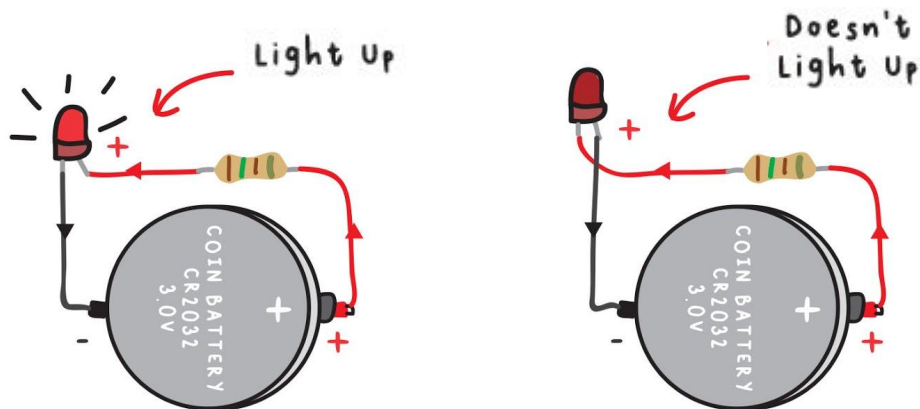


A "Toggle" azt jelenti, hogy váltani lehet az egyik állapotról a másikra. Ha az aktuális állapot be van kapcsolva, akkor az kikapcsol; és fordítva. Tehát, amikor egy LED-et ismételten bekapcsolunk, a LED villogni fog

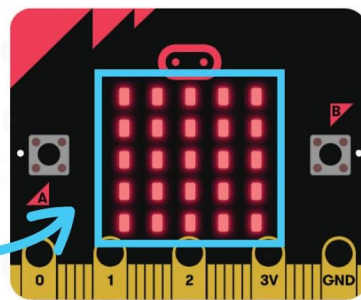
Érdekesség!



A fénykibocsátó dióda (LED) egy félvezető eszköz (egyik irányba vezeti/másik irányba nem vezeti az áramot). Két lába van, azaz pozitív és negatív. Ha a LED-et a megfelelő polaritással csatlakoztatják, és áram áramlik rajta, a LED fényt bocsát ki.



A micro: bitben használt elemek felületre szereléses technológián (SMT) alapulnak, és ezek nagyon aprók.



A micro: bit-en kívül egy másik SMT LED(41 db) is van az EDU: BIT kártyán. Látja őket?



youtu.be/qqBmvHD5bCw

Alkalmazás kihívás

Program EDU:BIT pontszámlálóként és időzítőként is működik, mint például a "Nyerni, Veszíteni vagy Rajzolni" és a "Színjátékban".

On start	Állítsa az A csapat változóját 0-ra Állítsa a B csapat változóját 0-ra
„A” gomb/ Sárga gomb	Az A-csapat értékét emelje meg 1-el. Mutassa meg az A-csapat aktuális eredményét
„B” gomb/ Kék gomb	A B-csapat értékét emelje meg 1-el. Mutassa meg a B-csapat aktuális eredményét
„A+B” gomb/ Sárga+Kék gombok	Görgessen az A és a B csapat eredményéhez
Rázáskor	Indítsa el az időzítőt 1 percre úgy, hogy kigyullad a zöld LED (30 másodpercre), majd a sárga LED (20 másodpercre) és végül a piros LED (10 másodpercre). Játssza le a "wawawaaa" dallamot, ha lejár az idő. Tíz alkalommal kapcsolja be a piros LED-et.

Itt egy tipp az Ön számára. Két változót kell elkészítenie, és el kell neveznie őket A és B csapatnak.



Kezdődhet a játék!

Win, Lose or Draw~



Hogyan játszunk:

- Osszuk el az osztályt 2 csapatra - A és B csapatra.
- Az A csapat egyik embere véletlenszerűen vesz ki egy kártyát. Miután elolvasta(magában) a szót a kártyán, rázza meg az EDU: BIT-et az időzítő elindításához (1 perc).
- Ekkor képeket rajzolhat a táblára, hogy a csapattagok kitalálják. Közben beszélgetni és mutogatni tilos!
- Egy pontot kap az A csapat (nyomja meg az A gombot vagy a sárga gombot), ha a csapat egyik tagja helyesen kitalálja a szót vagy a kifejezést, mielőtt az idő lejár.
- Ha az A csapat kudarcot vall, a B csapat megpróbálhat egy pontot "ellopni" azzal, hogy kitalálja a szót.
- Mindkét csapat felváltva rajzol és kitalál, a játék végéig.
- Az a csapat győz, aki a legtöbb pontot szerezte.

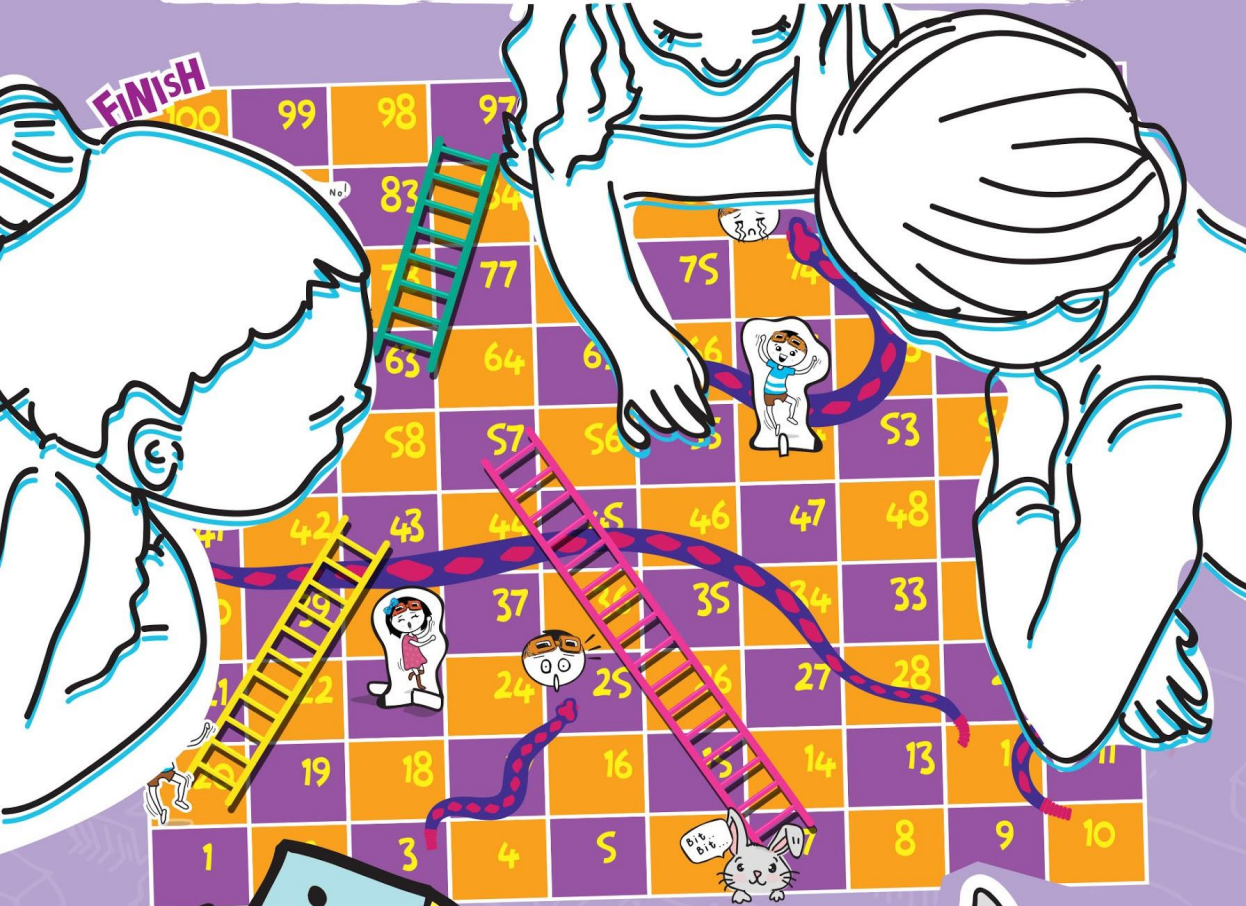
Megjegyzés!



Szkennelhet, hogy letölthessen nyomtatható kártyákat, javasolva kihívó szavakkal. Ha még nem rajzolt, kipróbálhatja a „Charade” alkalmazást. Ugyanazok a szabályok vonatkoznak, de a rajzolás helyett gesztusokat használunk, hogy a tagok kitalálják. Érezd jól magad!

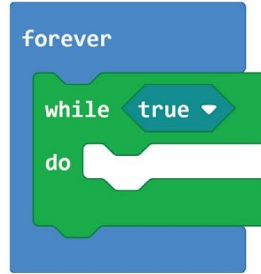
IR Digitális kocka.. Csináljuk

IR Bit

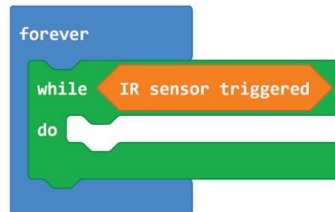
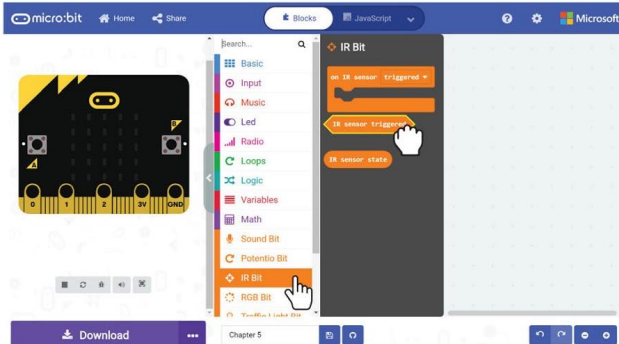


Gyerünk kódolni!

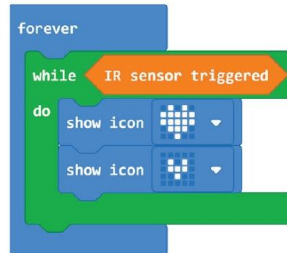
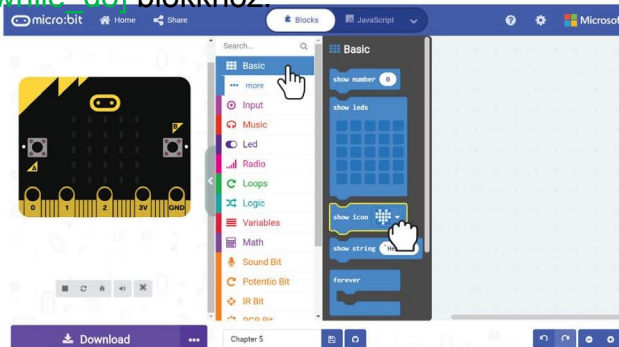
1.lépés: Hozzon létre egy új projektet a MakeCode Szerkesztőben, és adja hozzá az EDU: BIT kiterjesztést (lásd a 40. oldalon). Kattintson a **[Loops]** kategóriára, majd válassza a **[while_do]** blokkot. Helyezze a blokkot a **[forever]** nyílásba.



2.lépés Kattintson az **[IR Bit]** kategóriára, majd válassza az **[IR sensor triggered]** blokkot. Csatlakoztassa a blokkot a feltételes nyíláshoz a **[while_do]** blokkhoz.

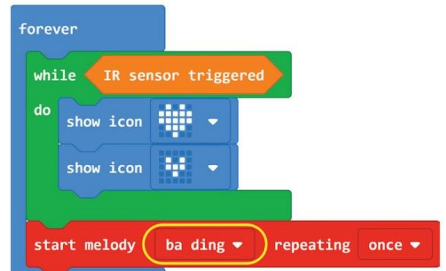


3.lépés Kattintson a **[Basic]** kategóriára, és adjon hozzá két **[show icon]** blokkot. Cserélje az egyik ikont egy "kis szívre". Csatlakoztassa mindkét blokkot a **[while_do]** blokkhoz.

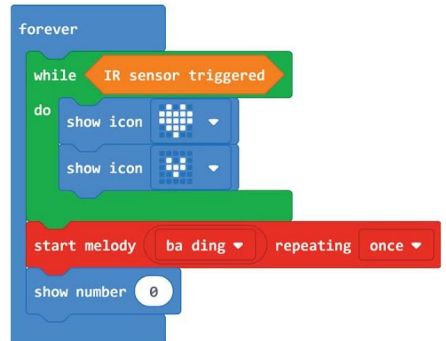
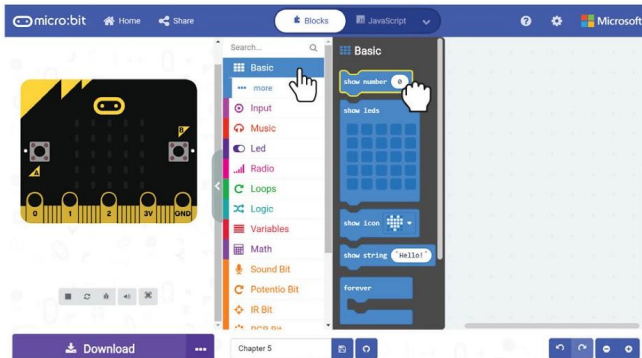




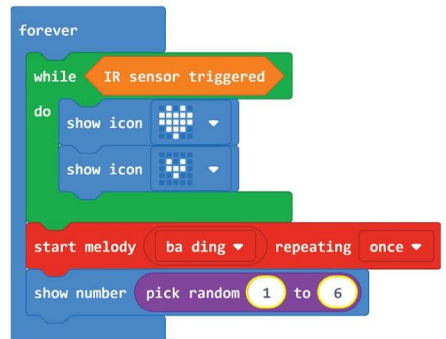
4.lépés Kattintson a [Music] kategóriára, majd válassza a [start melody _repeating_] blokkot. Változtassa a "dadadum" dallamot "ba ding" -re.



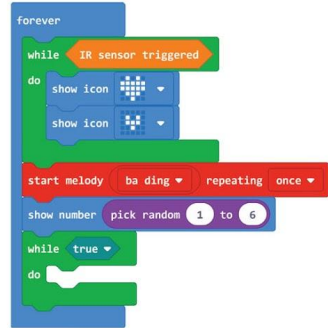
5.lépés Kattintson a [Basic] kategóriára, majd válassza a [show number] blokkot.



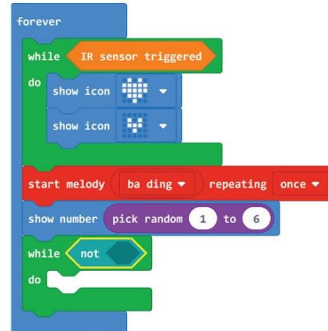
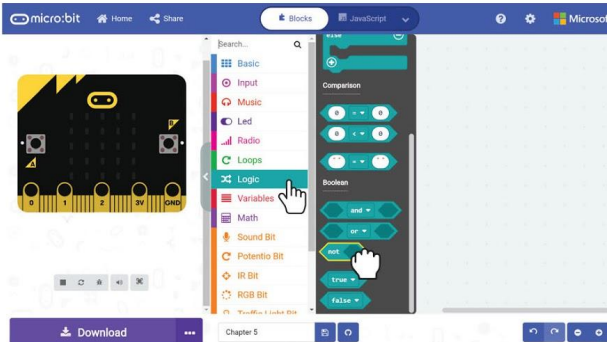
6. lépés Kattintson a [Math] kategóriára, majd válassza a [pick random _to_] blokkot. Állítsa a számokat 1-re és 6-ra.



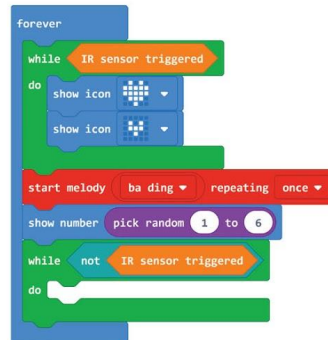
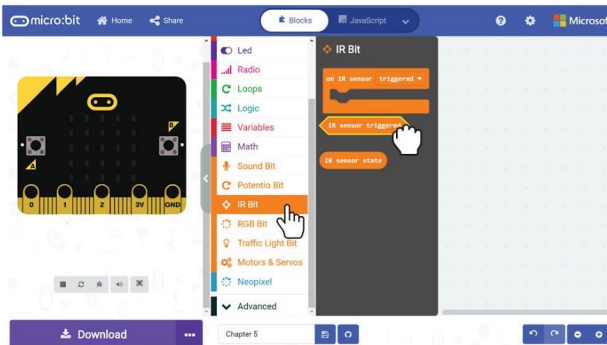
7.lépés Kattintson a [Loops] kategóriára és válassza a [while_do] blokkot.



8.lépés Kattintson a [Logic] kategóriára és válassza ki a [not] blokkot. Csatlakoztassa a blokkot a feltételes nyíláshoz a [while_do] blokkon.



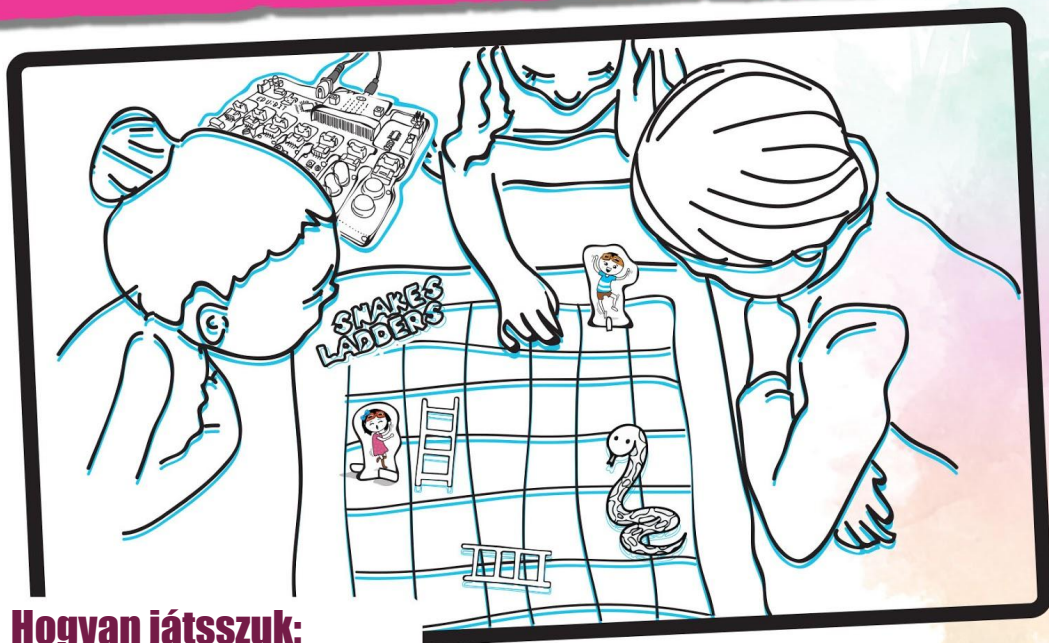
9.lépés Kattintson az [IR Bit] kategóriára és válassza ki az [IR sensor triggered] blokkot. Csatlakoztassa a blokkot a [not] blokk üres nyílásához.



10.lépés Töltse fel a kódot az EDU:BIT-re.

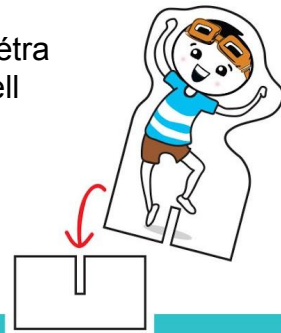
Kezdődhet a játék!

Kígyók és Létrák Játéka



Hogyan játszuk:

- Minden játékos válasszon egy karaktert, és helyezze a „Start Here” feliratú helyre.
- A játékosok felváltva "dobják el a kockát" - tegye a tenyerét az IR bit fölé. Amikor a dobogó szív animációt lát, vegye le a tenyerét.
- Mozgassa előre a karakterét a LED-mátrixon megjelenített számok szerint (1-6 között).
- Ha a karaktered a létra alján van, akkor lépkedhet fel a létra tetejére. Ha a karaktered egy kígyó fejére jut, akkor le kell csúsznia a kígyó farkának végére.
- Az első játékos, aki eléri a 100-at, az a győztes. Sok győzelmet!



Megjegyzés!

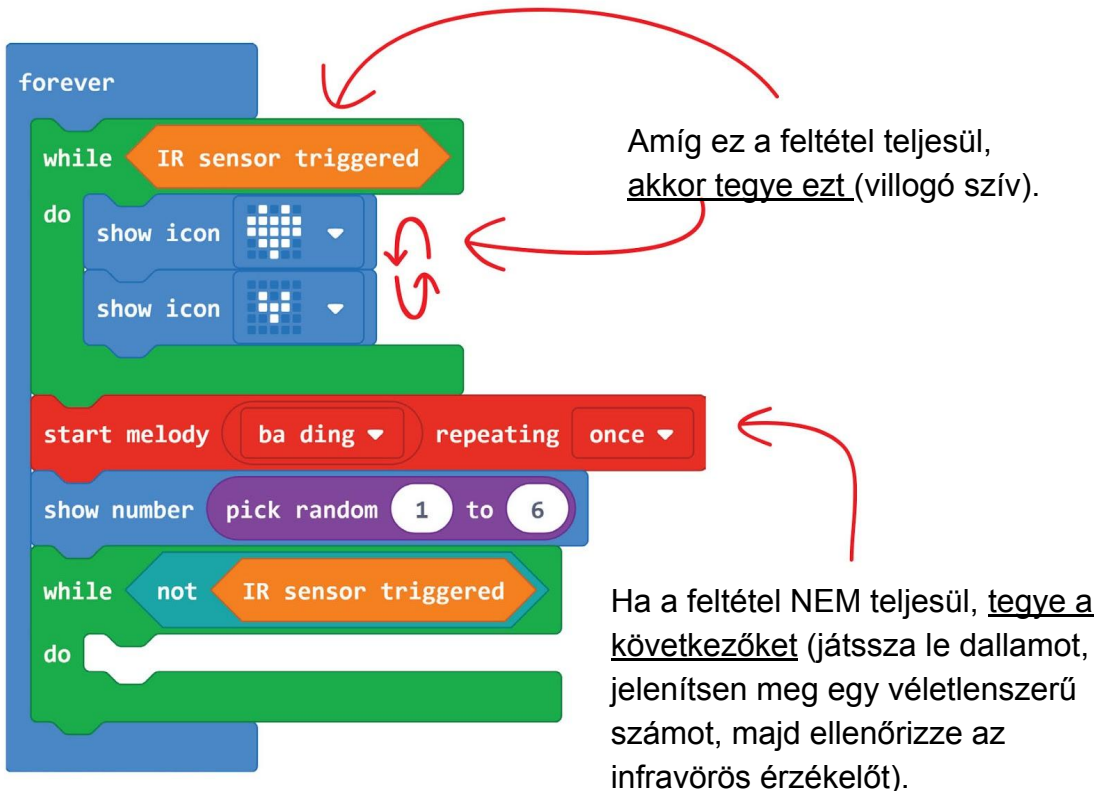
A kígyók és a létrák játéktábláját és karaktereit a dobozból lehet kivenni.

Értelmezzük a program működését



Korábban a `[while_do]` blokkot használtunk a `[Loops]` kategóriából kódunkban. Tudja, hogy meddig működnek a ciklusok?

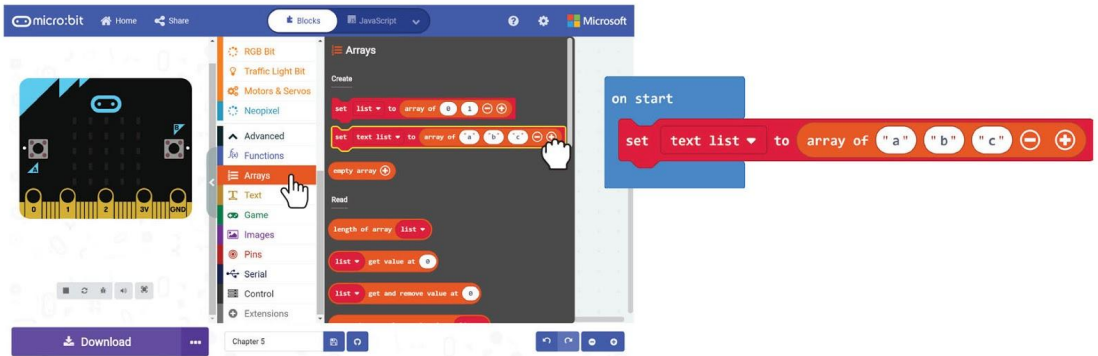
Amikor a program egy `[while_do]` blokkba kerül, ellenőrzi az állapotot. Amíg a feltétel teljesül (vagy igaz,) a program végrehajtja a kódblokk (oka) t a `[while_do]`. Ez tovább folytatódik, de amint a feltétel NEM teljesül (vagy hamis lesz), a program kilép a ciklusból és futtatja a következő kódblokkokat.



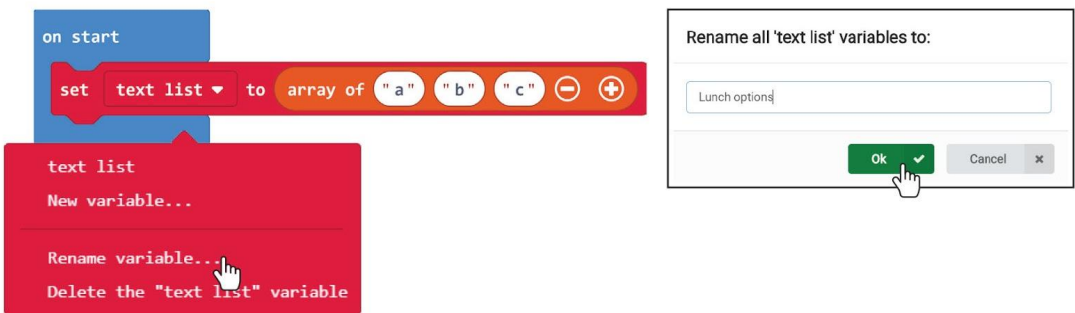


Az EDU: BIT digitális kockaként való felhasználása mellett módosíthatja a korábbi kódot is, ha több ugyanolyan csábitó lehetőséggel rendelkezik, és nem tudja eldönteni, hogy melyiket válassza, például - mit eszik ebédre? Yum Yum

11. lépés Kiklikeljen [Advanced]:[Arrays] kategóriára és válassza a [set text list to array of _ _ _] blokkot. Csatlakoztassa a blokkot a [Basic]: [on start] nyílásba.



12. lépés Kiklikeljen a [text list] blokkra és válassza a „Rename variable”-t. Az előugró ablakba írja be a 'Lunch options'-t és nyomja meg az Ok gombot.

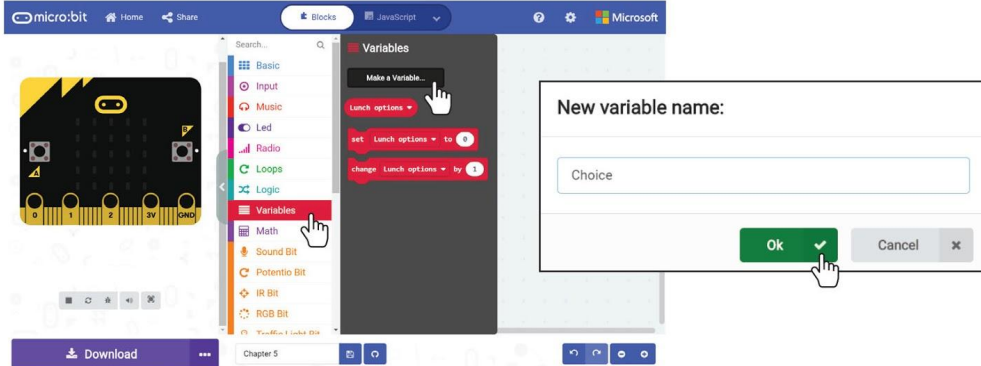


13. lépés Kattintson a tömb blokkra egyenként, és írjon be minden ebédbe egy ebéd opciót.

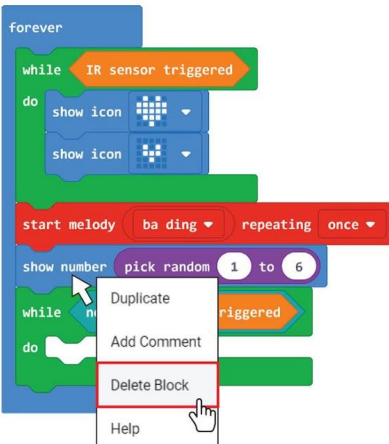


Kattintson erre a gombra további lehetőségek felvételéhez.

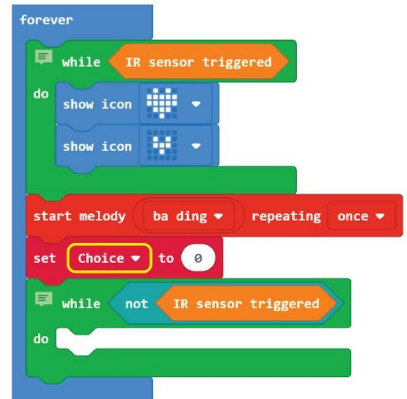
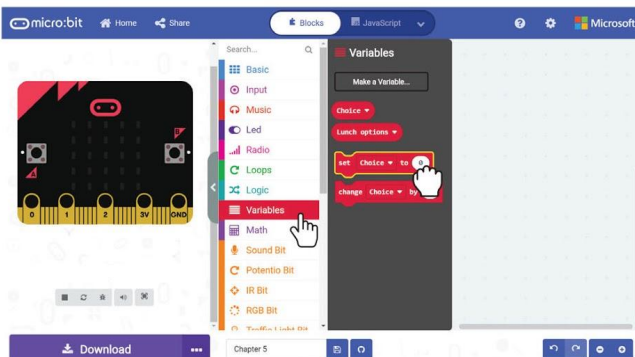
14.lépés Kattintson a **[Variable]** kategóriára és készítsen egy új változót 'Choice' néven.



15.lépés Kattintson a jobb gombbal a **[show number [pick random [pick random_to_]]]** blokkra, és válassza a 'Delete Blocks' lehetőséget.

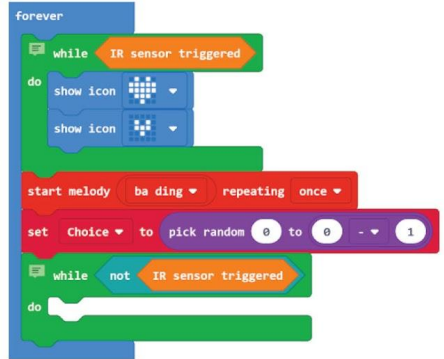
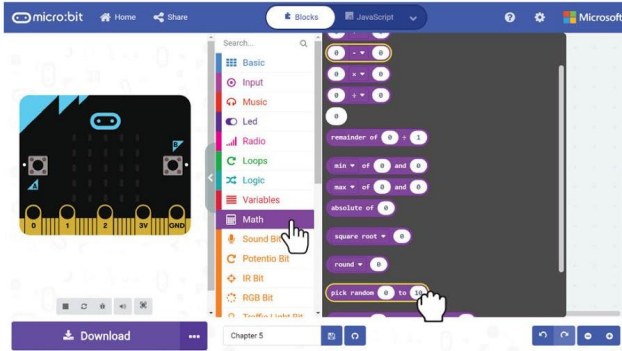


16.lépés Kattintson **[Variable]** kategóriára és válassza a **[set_to_]** blokkot. Helyezze a **[start melody_repeating_]** blokk és a **[while_do]** blokk közé. Változtassa meg a változót 'Choice' értékre.

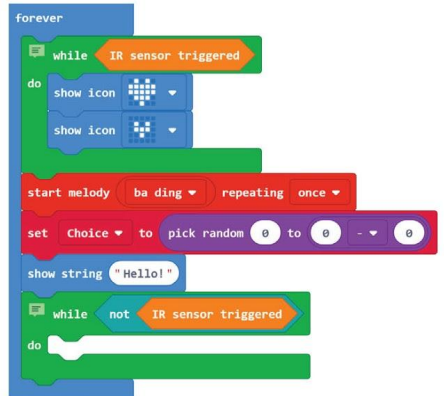
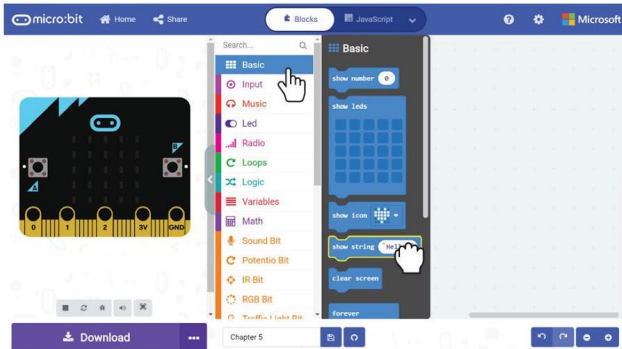




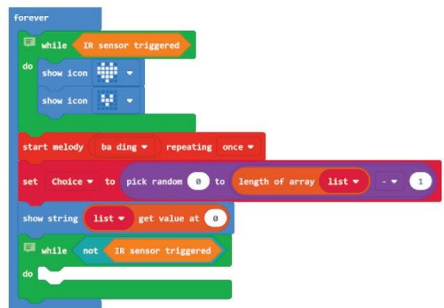
17. lépés Kattintson [Math] kategóriára és adja hozzá a [pick radom_to_] és [_] blokkokat a kódhoz. Változtassa meg az utolsó értéket 1-re.



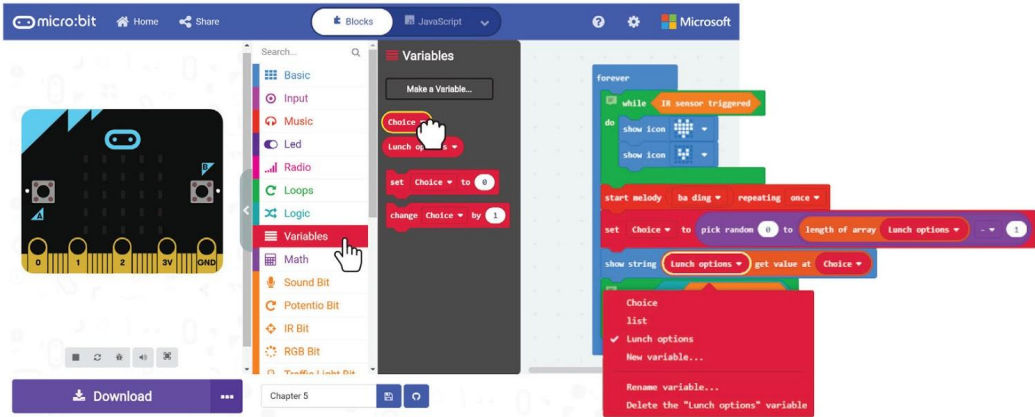
18. lépés Kattintson [Basic] blokkra és válassza a [show string] blokkot.



19. lépés Klickeljen [Advanced]:[Arrays] kategóriára és adja a [length of array_] és [get value at_] blokkokat a kódhoz.



20.lépés Kattintson a [list] elemre, és változtassa meg mindkét blokk esetében a "Lunch options" változót. Végül kattintson a [Variable] kategóriára, és válassza a [Choice] blokkot. Helyezze be a [_get value at_] blokk üres nyílásába.



Itt van a teljes "Mi van ebédre?" kód:

```
on start
  set Lunch options to array of "Fried rice" "Spaghetti" "Nasi lemak"

forever
  while IR sensor triggered
  do
    show icon
    show icon

  start melody ba ding repeating once

  set Choice to pick random 0 to length of array Lunch options

  show string Lunch options get value at Choice

  while not IR sensor triggered
  do
```

Itt természetesen más is szerepelhet!

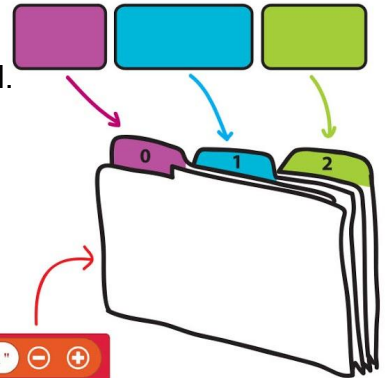


A következő alkalommal, amikor nem biztos benne, mit kell enni, hagyhatja, hogy az EDU: BIT az Ön számára döntsön az IR bit beindításával, majd a kezét elmozdítva tőle. Ezt a kódot módosíthatja, hogy segítsen dönteni arról, hogy melyik játékot játssza barátaival. Tudja, mit kell változtatnia ehhez?

Értelmezzük a program működését

Egy tömb a kapcsolódó változók listája vagy gyűjteménye. Gondolhatsz rá úgy, mint egy több szakaszos mappára, és minden szakasz egy darab információ tárolására szolgál. Egy tömböt használunk, így könnyen módosíthatjuk kódunkat, amikor elemeket kell hozzáadnunk vagy eltávolítanunk a listáról.

Ebben a kódban például létrehozott egy tömböt három elemmel, és "Lunch options" -nak nevezte el. Ezután egyszerűen szerkeszthetjük az egyes elemek által képviselt ételmet. Felvehetünk opció (ka) t is, vagy a + vagy - gombra kattintva csökkenthetjük a listában szereplő elemek számát.



```
on start
  set Lunch options to array of "Fried rice" "Spaghetti" "Nasi lemak"
```

```
forever
  while IR sensor triggered
  do
    show icon [grid icon]
    show icon [grid icon]
  start melody ba ding repeating once
  set Choice to pick random 0 to length of array Lunch options
  show string Lunch options get value at Choice
  while not IR sensor triggered
  do
```

Ha ez a feltétel nem teljesül, az EDU: BIT a következőket fogja tenni:

egyszer lejátssza a „ba ding” dallamot,

véletlenszerűen válasszon egy elemet az ebéd lehetőségek listájáról, és

mutassa meg a LED kijelzőn.

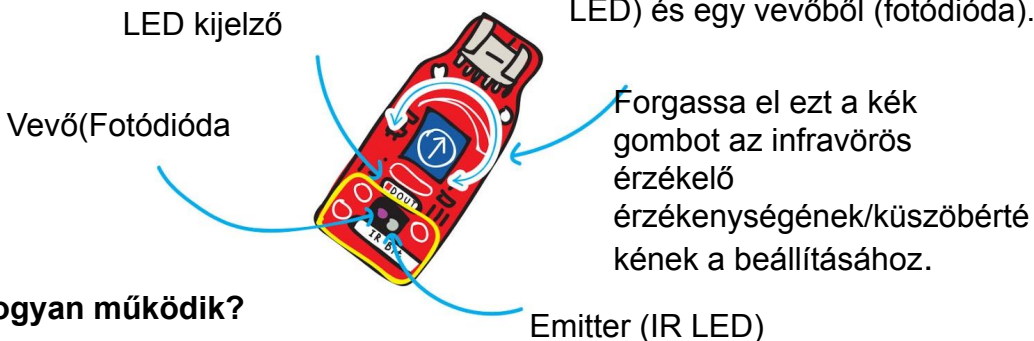


A programozás során az 1 helyett 0-tól kezdjük a számlálást. Ezért a "Fried rice" a lista 0-as számán van, és az utolsó "Nasi lemak" elem a 2-es indexszámnál van, még akkor is, ha az a harmadik tétel.

Érdekesség!



Az **infravörös (IR)** érzékelő egy elektronikus eszköz, amelyet általában az akadályok felismerésére használnak. Az IR érzékelő két részből áll - egy emitterből (IR LED) és egy vevőből (fotodióda).

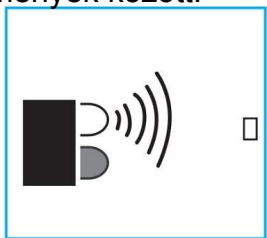


Hogyan működik?

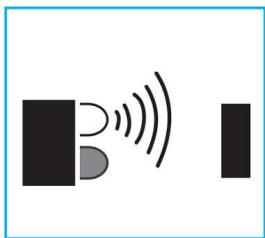
Az IR LED fényt bocsát ki, amely visszatükröződik a vevőn, ha tárgyat helyeznek az érzékelő elé. Az IR bit "aktiválódik", ha a visszavert fény mennyisége meghaladja a küszöbértéket. Bekapcsoláskor az IR bit LED-jelzőfénye kigyullad.

Ha nincs tárgy vagy a tárgy túl messze van, akkor nagyon kevés vagy egyáltalán nincs infravörös fény visszaverve a vevőre. Ezért az IR bit nem indul el.

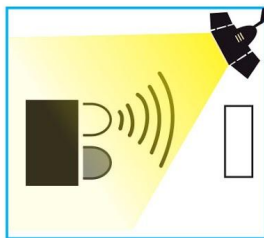
Az infravörös érzékelő azonban nem működik a várt módon a következő körülmények között:



Az objektum túl kicsi



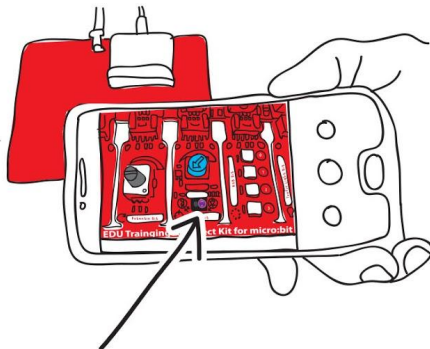
Az objektum sötét színű felülettel rendelkezik



Fény zavarása

Tudtad?

Az infravörös fény szabad szemmel láthatatlan. Az infravörös fényt azonban megtekintheti azáltal, hogy egyszerűen az infravörös LED-et egy telefonkamerán nézi.

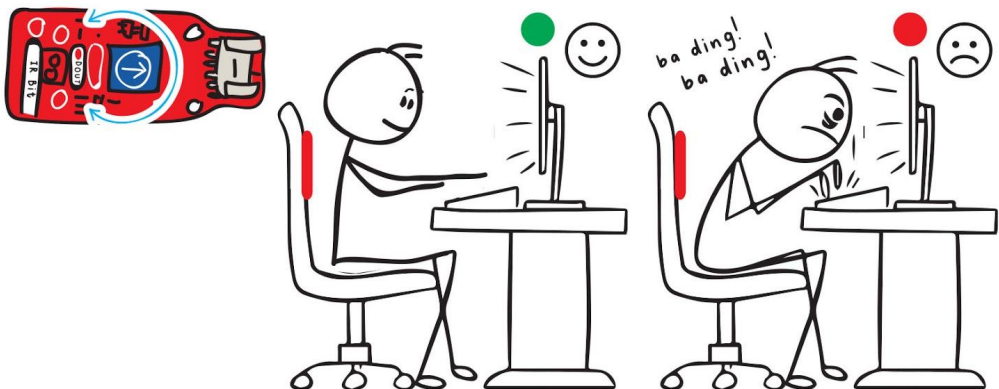


Alkalmazás kihívás

EDU:BIT program lecsúszásgátló detektor	
On start	Görgetési emlékeztető - Vegye figyelembe a testtartását
On IR triggered	Mutasson egy mosolygó arcot a LED mátrix kijelzőjén, és világítson meg zöld LED-et.
On IR NOT triggered	Indítsa el a „ba ding” dallam egyszeri ismétlését. Mutasson egy szomorú arcot a LED mátrix kijelzőjén, és villogjon a piros LED.

Hogyan működik:

Csatlakoztassa az EDU: BIT-et a szék hátuljához, az ábra szerint. Üljen kényelmesen, jó testtartással. Addig állítsa be az IR bit kék gombját, amíg a LED jelzőfény kigyullad (az IR bit érzékeli a hátát). Ezt a folyamatot kalibrálásnak nevezzük.

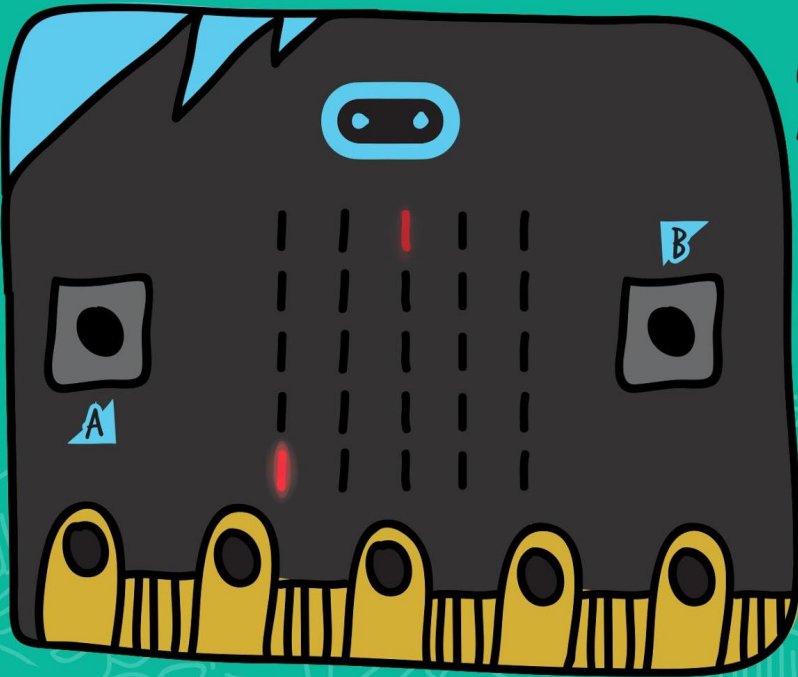


Ha más színű felsőt visel, akkor újra kell kalibrálnia az IR bit-et. Tudod miért?



Hé, te vagy az!

Potentio Bit

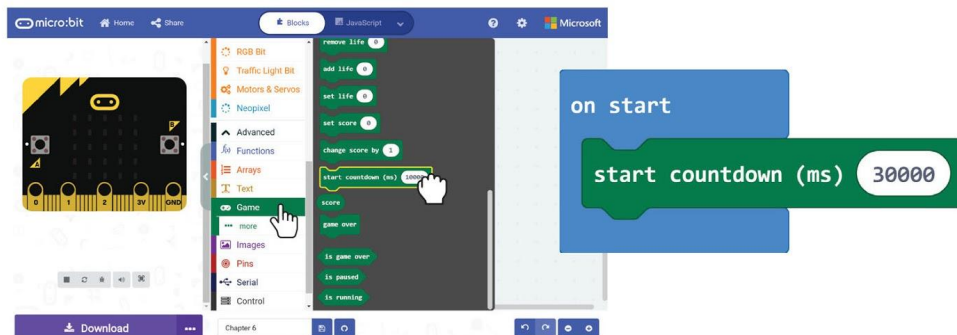


link.cytron.io/edubit-chapter-6

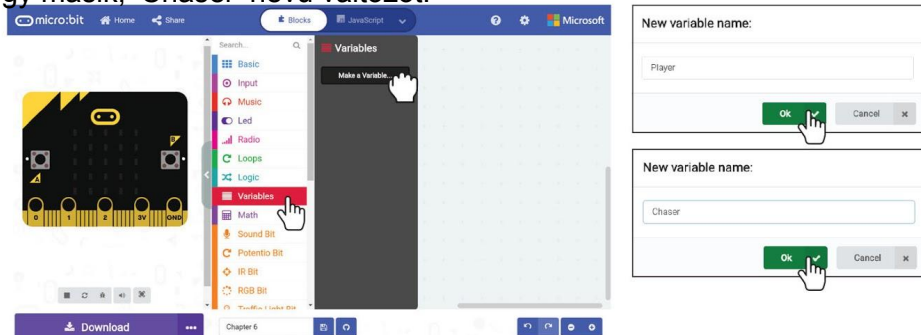


Gyerünk kódolni!

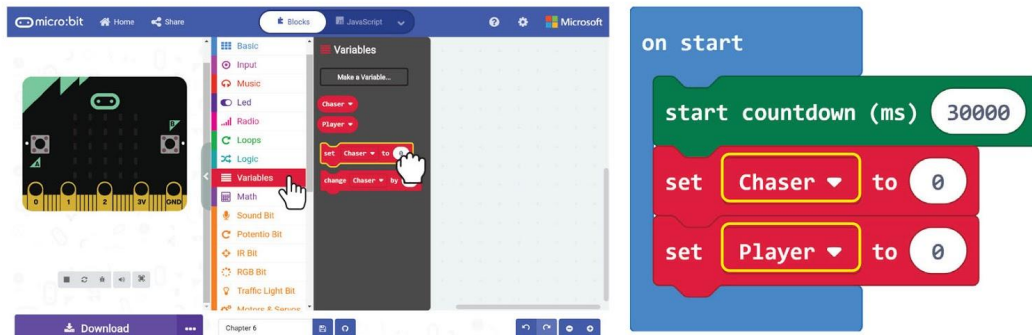
1.lépés Készítsen új projektet a MakeCode Szerkesztőben, és adjon hozzá egy EDU: BIT kiterjesztést (ajánlhatja a 40. oldalt). Kattintson az [Advanced], majd a [Game] kategóriára. Válassza a [start countdown (ms)] blokkot, kattintson rá [on start] blokkra, és változtassa meg az értéket 30000 értékre.



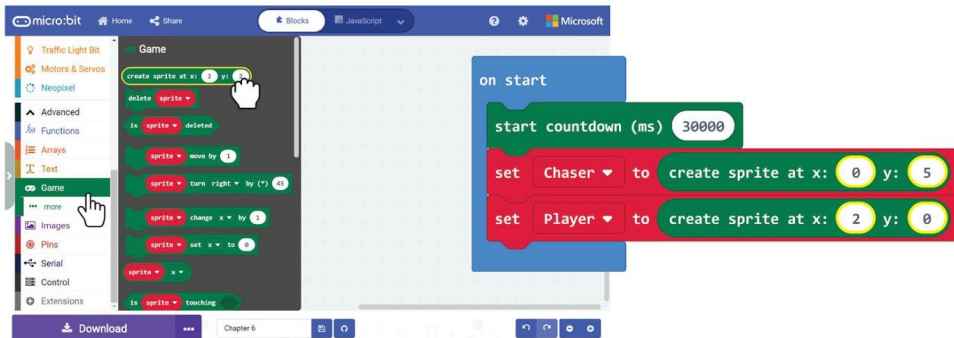
2.lépés Kattintson a [Variables] kategóriára, majd a [Make a Variables] elemre. Írja be a „Player” elemet a felbukkanó ablakba, majd kattintson az OK gombra. Hozzon létre egy másik, 'Chaser' nevű változót.



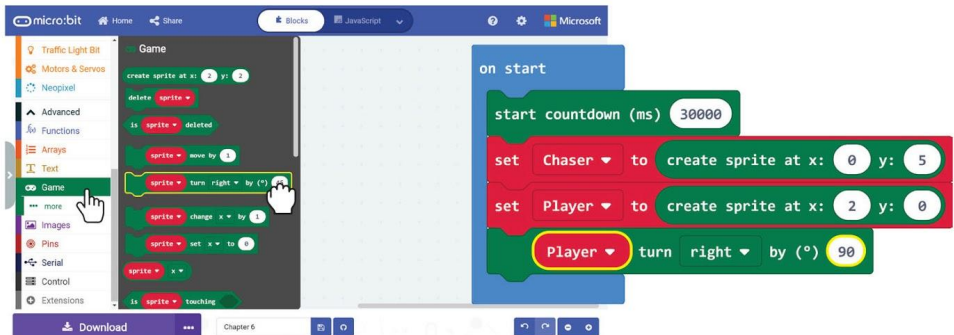
3.lépés Kattintson a [Variables] kategóriára, majd válassza a [set_to_] blokkot. Másolja a [set_to_] blokkot, és illessze be mindkét blokkot a [start] blokkba. Állítsa be a 'Chaser' és a 'Player' változókat.



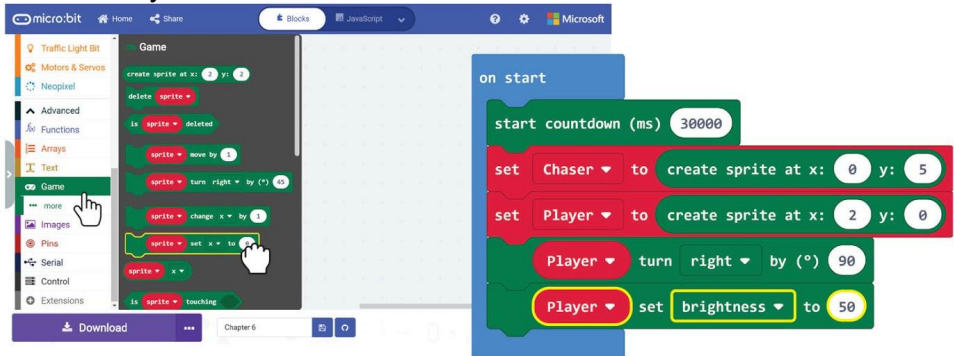
4.lépés Kattintson az [Advanced], majd a [Game] kategóriára. Válassza a [create sprite x: _y: _] blokkban lehetőséget. Másolja és illessze be a blokkokat a [set_to_] blokkokba. Változtassa meg az értékeket x: 0 y: 5 -hez 'Chaser' és x: 2 y: 0-ra 'Player' -hez.



5.lépés Kattintson [Advanced]:[Game] kategóriára és válassza a [turn_by(°)] blokkot. Helyezze a blokkot a [on start] nyílásba. Válassza ki a „Player” változót, és állítsa a fokot 90-re.

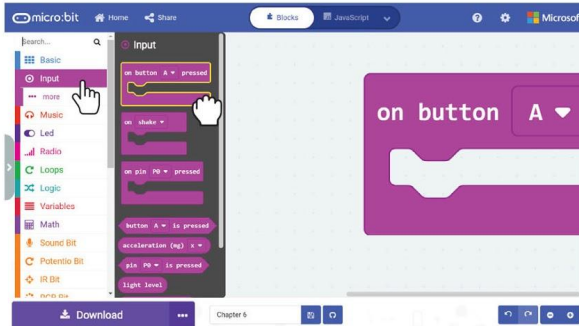


6.lépés Kattintson [Advanced]:[Game] kategóriára és válassza a [set_to_] blokkot. Válassza ki a 'Player' változót, változtassa meg az 'x' értéket 'brightness' értékre, és állítsa a fényerőt 50-re.

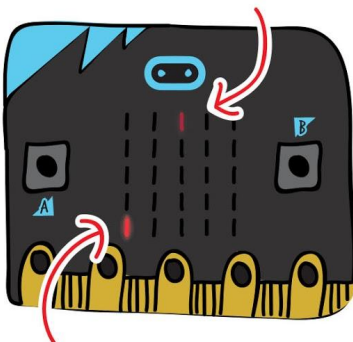
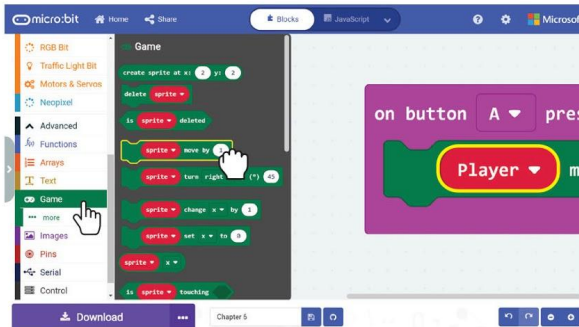




7.lépés Kattintson [Input] kategóriára és válassza a [on button_pressed] blokkot. Másolja a blokkot, és válassza ki a 'B' gombot a második [on button_pressed] blokkban.



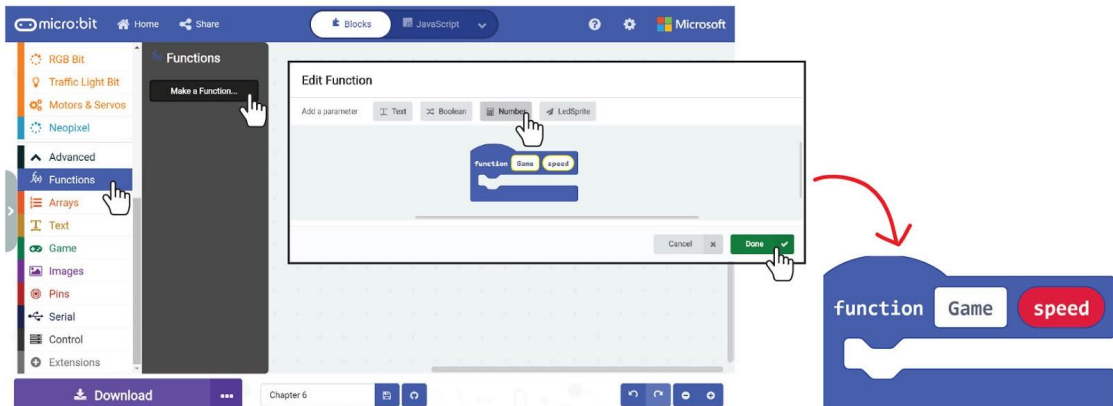
8.lépés Kattintson a [Game] kategóriára és válassza a [_move by_] blokkot. Másolja és helyezze el a blokkokat az [on button_pressed] nyílásokba. Válassza ki a 'Player' változót mindkét blokkhoz, és adja meg az értékeket -1-ig (az A gomb megnyomásakor) és 1-ig (a B gomb megnyomásakor).



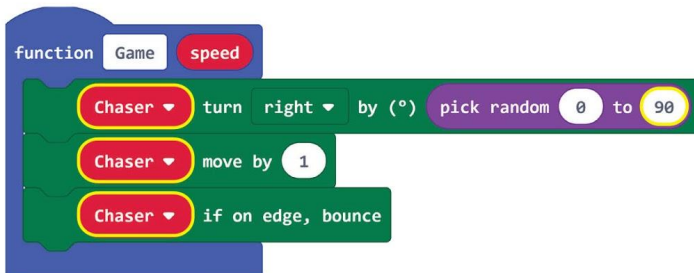
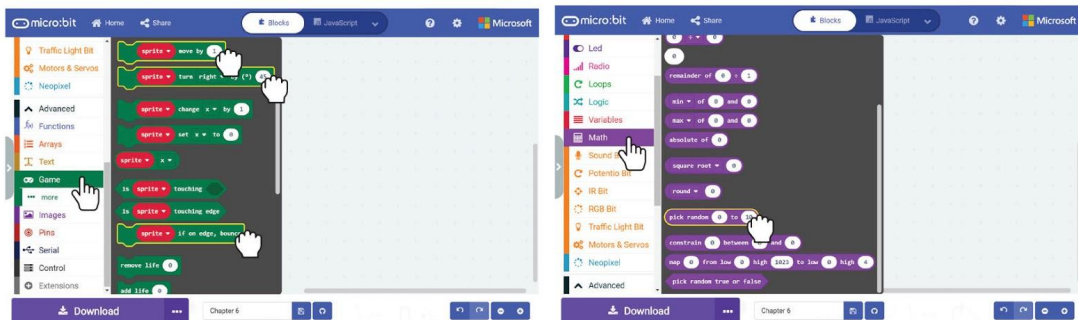
EDU:BIT-re töltsse fel akódot. Amikor megnyomja a kék gombot (B gomb), észreveszi, hogy a tompított LED fény lefelé mozog? Ez a játékos lelkesedése! A sprite olyan, mint egy "kis LED-es lény", amelyet irányíthat. Mi történik, ha megnyomja a sárga gombot (A gomb)?

6. fejezet : Poentio Bit

9. lépés Kattintson az **[Advanced]** kategóriára és válassza a **[Functions]** kategóriát, majd kattintson a **[Make a Function...]** gombra. A Funkció szerkesztése ablakban nevezze át a „doSomething” -et „Game” -re. Ezután kattintson a **[Number]** gombra egy paraméter hozzáadásához és a „num” átnevezéséhez a „speed” a funkcióblokkban. Ezután kattintson a 'Done' gombra.

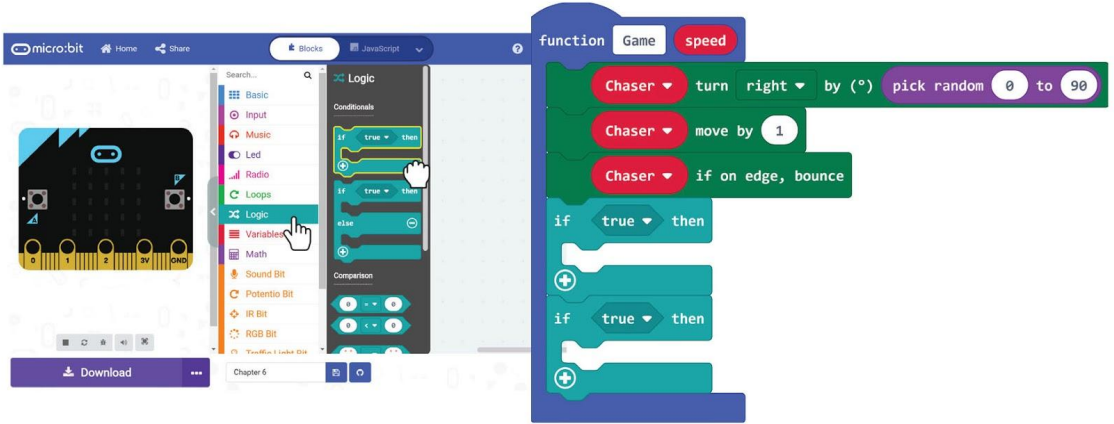


10. lépés Folytassa a kód létrehozását az **[Advanced]: [Game]** és **[Math]** kategóriákból származó blokkok hozzáadásával, az alábbiak szerint. Ne felejtse el a változót 'Chaser' -re, az értéket 90-re változtatni.

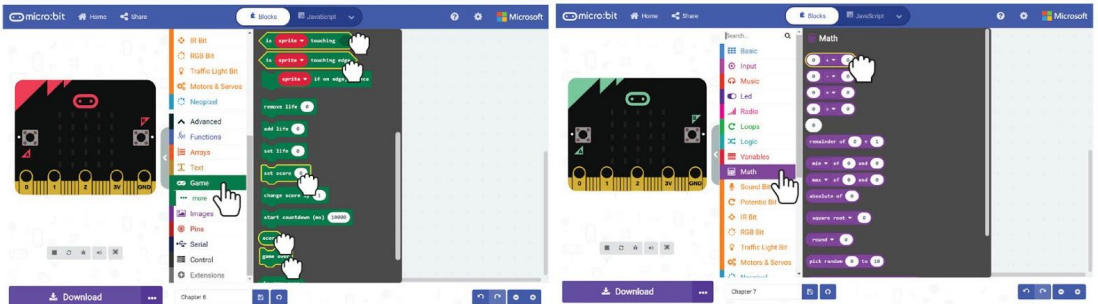




11. lépés Adjon hozzá két [if-then] blokkot a [Logika] kategóriából a kódjához.



12.lépés Folytassa a kód létrehozását az [Advanced]: [Game] és [Math] kategóriákból származó blokkok hozzáadásával, az alábbiak szerint.



```

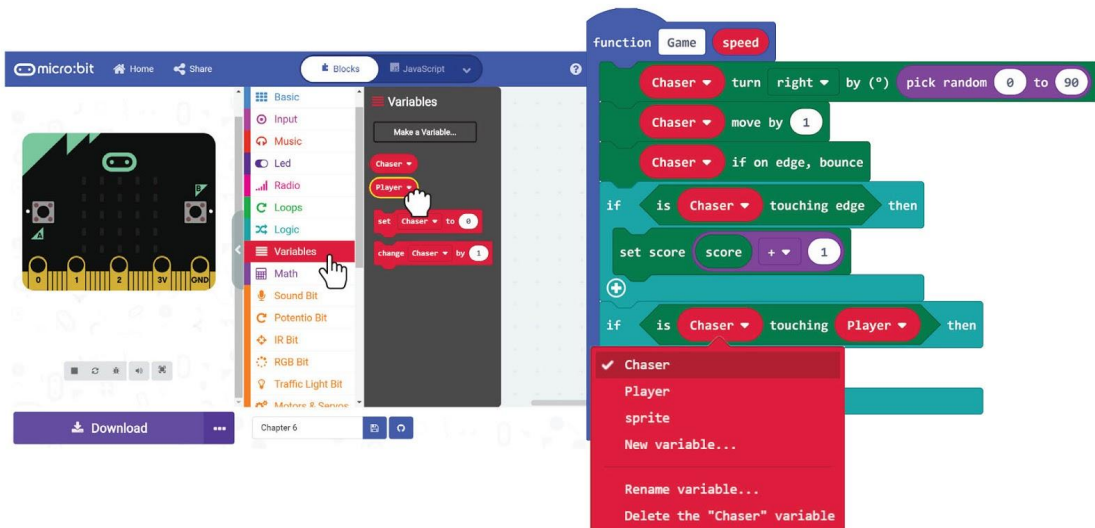
function Game speed
  Chaser turn right by (°) pick random 0 to 90
  Chaser move by 1
  Chaser if on edge, bounce
  if is sprite touching edge then
    set score score + 1
  if is sprite touching then
    game over
  
```

Folytasd!

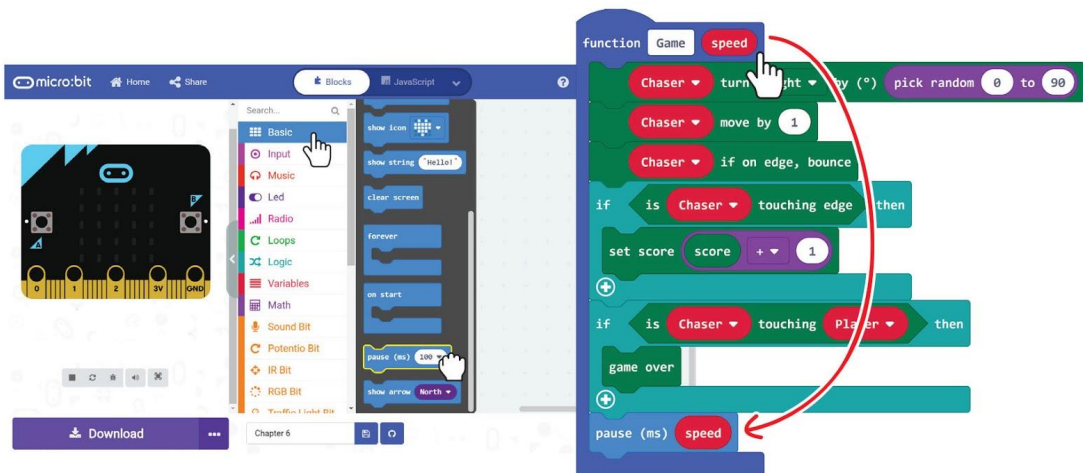


6. fejezet : Poentio Bit

13.lépés Cserélje ki mindkét [sprite] blokkot 'Chaser' -re kattintással, majd a 'Chaser' kiválasztásával. Kattintson a [Variables] kategóriára, és válassza a [Player] blokkot. Helyezze rá az [is_touching_] blokk üres nyílásába.



14.lépés Kattintson az [Basic] kategóriára, és válassza a [pause (ms)] blokkot. Adja hozzá a kódjához. Kattintson a [speed] gombra a funkcióblokkban, tartsa lenyomva és húzza az [pause (ms)] blokk üres nyílásába.



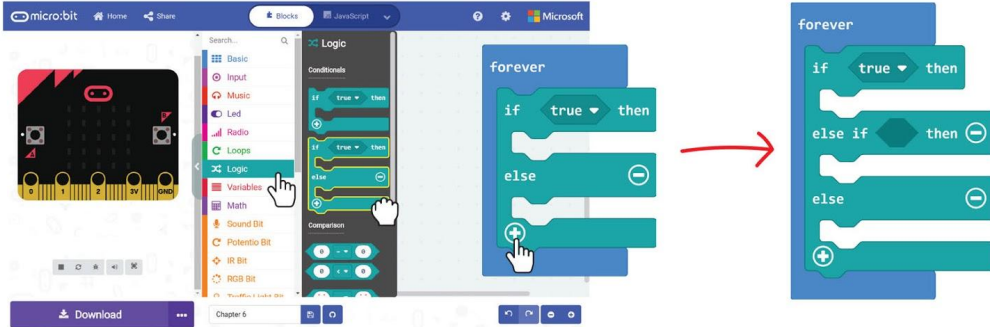
Izgalmassá tudja tenni a játékot, ha dallamot játszik le, amikor a Chaser megérinti a lejátszót. Meg tudja találni, melyik blokkot kell hozzáadni, és hová kell tenni?



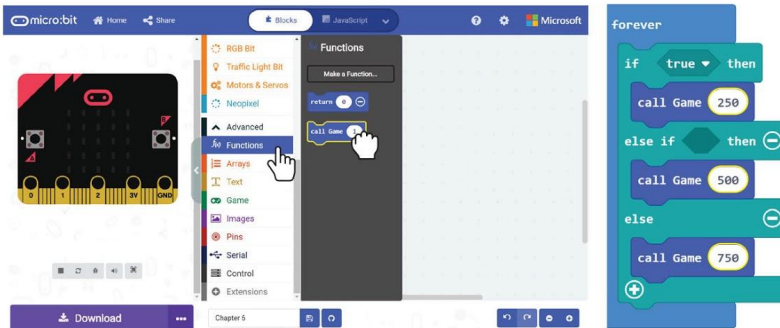


Most adjunk hozzá különböző nehézségi szinteket a játékhoz

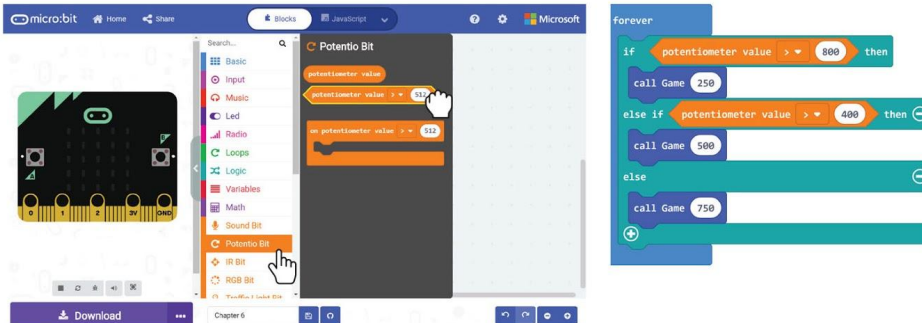
15.lépés Kattintson a [Logic] kategóriára, és válassza az [if-then-else] blokkot. Helyezze a blokkot a [forever] nyílásba. Kattintson a (+) gombra további feltétel hozzáadásához.



16.lépés Kattintson a [Functions] kategóriára és válassza a [call Game_] blokkot. Másolja a blokkot, és rögzítse a blokkokat az összes [if-then-else] slothoz. Változtassa meg a [call Game_] blokkot 250, 500 és 750 értékre.



17.lépés Kattints a [Potentiometer Bit] kategóriára és válassza a [potentiometer value>_] blokkot. Másolja és helyezze el a blokkokat az [if-then-else] blokk feltétel részeiben. Állítsa az értéket 800-ra az első blokkban és 400-ra a második blokkban.



Itt a teljes kód:

Mutasson egy animációt, majd indítson el egy 30 másodperces visszaszámlálót.

Hozzon létre egy Chaser sprite-t és egy Player sprite-t a megadott koordinátákon.

A Player sprite fordulással megkülönbözteti a Chaser sprite-től.

A Dim Player sprite megkülönbözteti a Chaser sprite-től.

```

on start
  start countdown (ms) 30000
  set Chaser to create sprite at x: 0 y: 5
  set Player to create sprite at x: 2 y: 0
  Player turn right by (°) 90
  Player set brightness to 50

on button A pressed
  Player move by -1

on button B pressed
  Player move by 1
  
```

Játék funkció a Chaser sprite vezérléséhez

Állítsa a Chaser sprite-t úgy, hogy véletlenszerűen 0 és 90 fok közötti szögben forduljon jobbra, majd 1 lépéssel mozogjon. Ha a Chaser sprite megérinti a szélét, akkor visszapattan.

Minden alkalommal, amikor a Chaser sprite megérinti a szélét, frissítse a "pontszám" változót, 1 pont hozzáadásával.

Ha a Chaser sprite megérinti a Player sprite játékot, játssza le a dallamot wawawawaaa, majd jelenítse meg a Game Over animációt és az eredményt.

Szüneteltesse a programot az ezredmásodperc számához a "speed" változó értéke szerint.

```

function Game speed
  Chaser turn right by (°) pick random 0 to 90
  Chaser move by 1
  Chaser if on edge, bounce
  if is Chaser touching edge then
    set score score + 1
  if is Chaser touching Player then
    start melody wawawawaaa repeating once
    game over
  pause (ms) speed
  
```

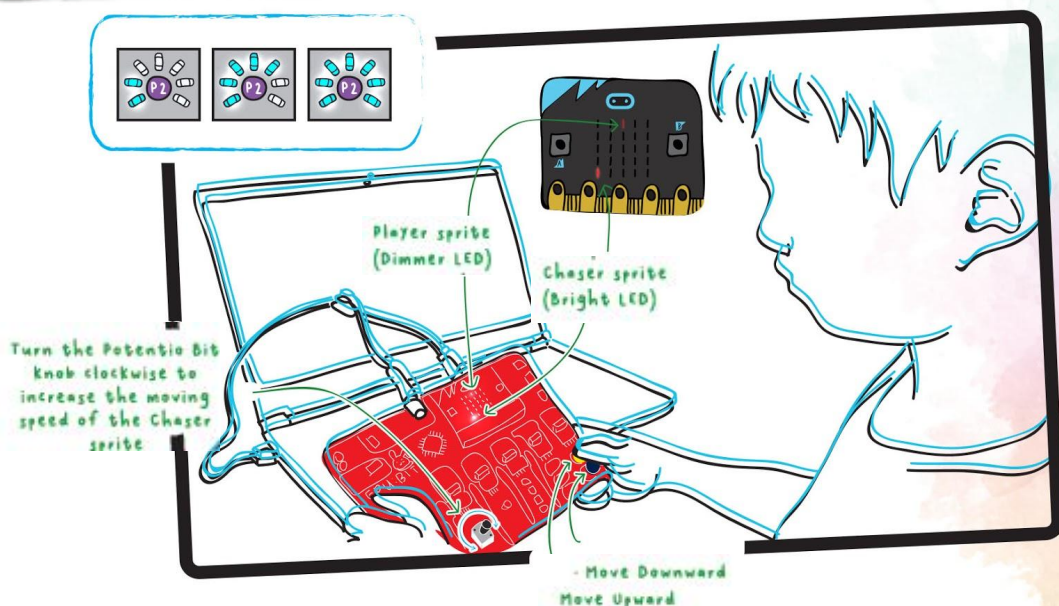
Mindig ellenőrizze a Potentio Bit olvasását.
 IF/Ha a potenciométer értéke nagyobb, mint 800, akkor hívja a "Game" funkciót és állítsa a "sebesség" változót 250 ms-ra.
 ELSE IF/Különb Ha a potenciométer értéke nagyobb, mint 400, akkor hívja a "Game" funkciót, és állítsa a "sebesség" változót 500 ms-ra.
 ELSE/Különbén hívja a "Game" funkciót, és állítsa a "sebesség" változót 750 ms-ra.

```

forever
  if potentiometer value > 800 then
    call Game 250
  else if potentiometer value > 400 then
    call Game 500
  else
    call Game 750
  
```

18.lépés Töltse fel a teljes kódot az EDU:BIT-re, és érezd jól magad. Irány játszani! a barátaiddal.

Játszunk!



Hogyan kell játszani:

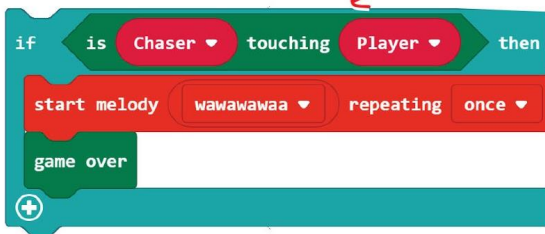
- Bekapcsolt állapotban a Chaser sprite véletlenszerűen mozog.
- Mozgassa a Player sprite-t felfelé vagy lefelé, hogy elkerülje a Chaser-sprite-t. Nyomja meg a sárga gombot (A gomb), hogy felfelé mozogjon, a kék gombot (B gomb), hogy lefelé mozogjon.
- A játék vége, ha a Chaser „megérinti” a lejátszót, vagy ha a 30 másodperces határidő lejárt.
- Minden alkalommal, amikor a Chaser sprite "megérinti" a szélét, 1 pontot szerez. A legmagasabb pontszámú játékos a győztes! Sok sikert!

Tipp!

- # 1 Ha magasabb pontszámot szeretne elérni a 30 másodperces időn belül, akkor növelheti a Chaser sprite sebességét, hogyha gyakrabban „megérinti” a szélét.
- # 2 A játék vége után az A + B gombokat egyszerre megnyomhatja egy új játék elindításához. Ez a [Game] blokkok beépített funkciója.

Értelmezzük a program működését

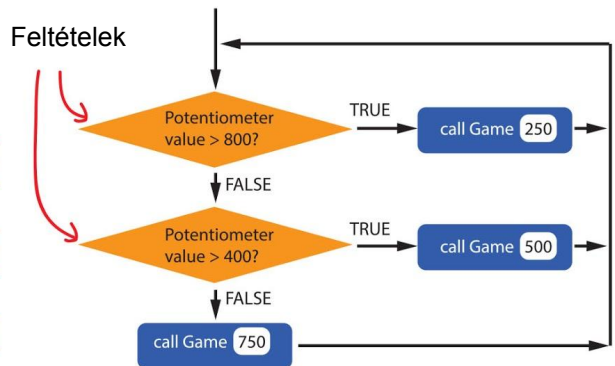
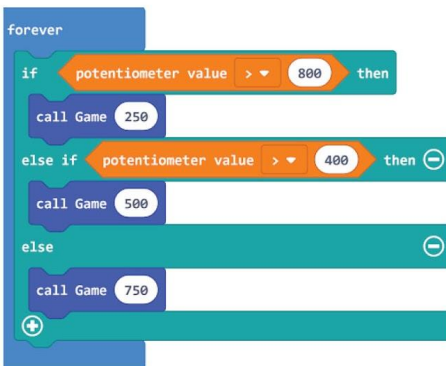
A programozásban feltételes ha utasításokat használunk a döntések meghozatalához. A MakeCode-ban [if-then] vagy [if-then-else] feltételes blokkokat használunk a [Logic] kategóriából a feltétel kialakításához. A program ellenőrzi a feltételt, és ha igaz, akkor végrehajtja a kódot a feltételes mondatban. Másként ha hamis, akkor a következő kódblokkba lép.



ha ez a feltétel teljesül (azaz a Chaser sprite megérinti a Player sprite-t)
akkor ezt tegye (játssza le a dallamot wawawawaa, majd jelenítse meg a „game over” animációt).

Ha több feltételünk van, a program sorrendben értékeli a feltételeket, felülről lefelé, és végrehajtja az első feltétel megfelelő kódját, amely igaznak bizonyul. Így minél magasabban helyezkedik, annál nagyobb prioritást élvez az alábbiakhoz képest.

Például, ez a kód a játékban meghatározza a Chaser sprite mozgási sebességét, összehasonlítva a potenciométer értékét az előre beállított küszöbértékekkel.



ha a potenciométer értéke > 800, hívja a Game funkciót (250 ms-ra állítható sebességgel),

különben ha a potenciométer értéke > 400, akkor hívja a Game funkciót (sebesség = 500 ms),

különben hívja a Game funkciót (sebesség = 750 ms).

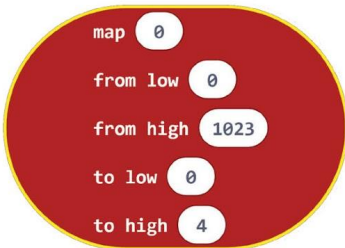


FEDEZZEN FEL TOVÁBBI BLOKKOKAT

1 Használjon egy [Basic]: [show number] blokkot a [Potentio Bit]: [potentiometer value] blokkal az aktuális potenciométer értékének olvasásához és megjelenítéséhez.

show number potentiometer value

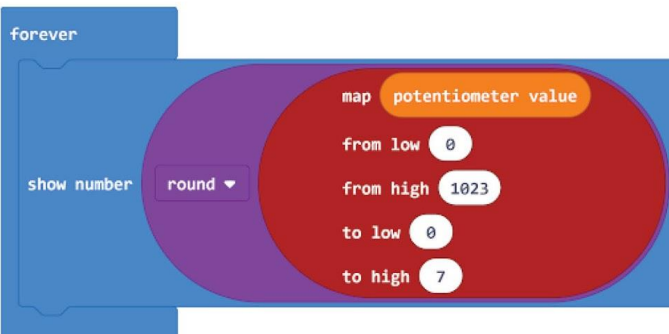
2 A potenciométer 0 és 1023 közötti értéket ad vissza. Használhatja a [map_from low_from high_to low_to high_] blokkot az [Advanced]: [Pins] kategóriából, hogy az olvasást egy másik, értelmesebb tartományba rendelje.



3 A leképezési blokk egy tizedes pontot tartalmazó számot ad vissza (például 1.68, 3.998). A szám kerekítéséhez használja a [Math] kategóriából a [round_] blokkot.

round 0

Íme egy példakód a Potentio Bit leolvasásának 0-tól 7-ig terjedő tartományba való leképezéséhez. Az értéket felfelé kerekítjük, és megjelenik a LED-mátrixon.



Ne felejtse el bekapcsolni az EDU: BIT-et a pontos beolvasáshoz.



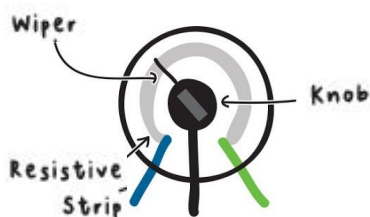
Vicces tény!



A potenciométerek, más néven pot(ok), változó ellenállások, amelyek ellenállása könnyen szabályozható egy gomb vagy egy csúszka segítségével.



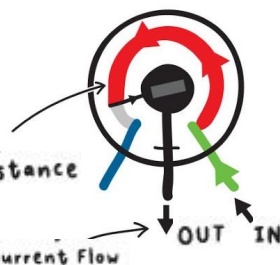
Ha 10 000 Ω potenciométered van, akkor az ablaktörő helyzetének megváltoztatásával 0 és 10 000 Ω közötti ellenállási értéket kaphatsz.



Less Resistance



More Resistance



Indicate Current Flow

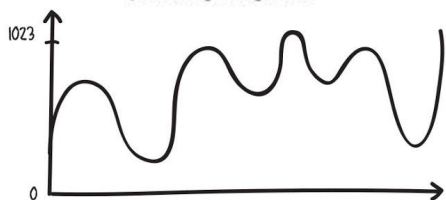


Néhány fontos alkalmazás:

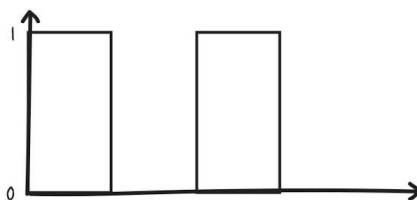
- A hangszóró hangereje
- Rádió frekvenciaszabályozása
- A vízmelegítő hőmérsékletének szabályozása

Az EDU: BIT potenciométere egy analóg bemeneti eszköz típusa. Az elektromos potenciált méri, és a mért feszültséget (0 V - 3,3 V között) 0 és 1023 közötti egész értékre konvertálja.

Analog Signal



Digital Signal



Alkalmazás kihívás

Az EDU:BIT egy időzítő legyen. A Potentio Bit segítségével állítsa be az időtartamot (0 és 60 másodperc között), az A gombot az időzítő aktiválásához és a B gombot a visszaállításhoz

On start	Állítsa az üzemmódot 0-ra
A gomb/ Sárga gomb	Állítsa az üzemmódot 1-re Állítsa a kezdési időt futási időre Mutasson egy mosolygó arc ikont
B gomb/ Kék gomb	Állítsa az üzemmódot 0-ra
Forever	Mindig ellenőrizze a módot <ul style="list-style-type: none">• IF Mode = 0, akkor állítsa az Időtartamot a potetiometer leolvasásának lekerekített értékére, leképezve az alacsony 0 és a magas 60 értékre, és az Időtartamot jelenítse meg a LED mátrixon.• ELSE IF mód = 1, ellenőrizze, hogy (futási idő - kezdési idő) > (Időtartam x 1000). Ha igaz, játssza le a dallamot wawawaaa, majd állítsa a módot 2-re.• Else, mutasson szomorú arcot.

Itt van néhány tipp neked:

Tipp #1 Három változót kell létrehoznia: Üzemmód, Indítási idő és Időtartam

Tipp #2 A futási idő (ms) blokk az [Input] kategóriából van.

Tipp #3 Használja ezt a feltételes kijelentést, hogy ellenőrizze lejárt-e az idő.



running time (ms)

- ▾

Start Time ▾

≥ ▾

Duration ▾

x ▾

1000

Tapsot kérünk!

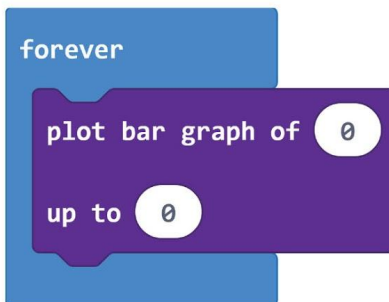
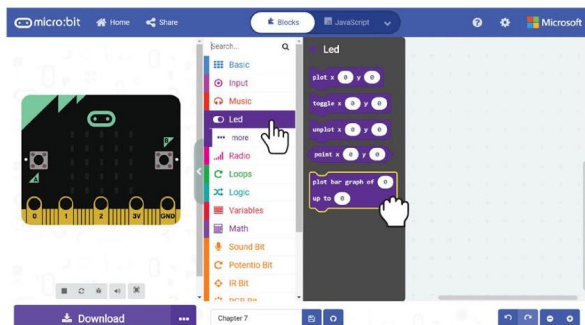
Sound Bit



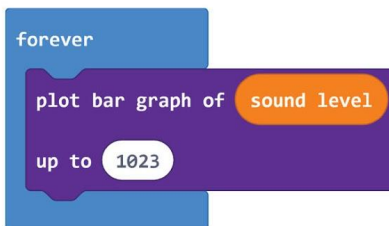
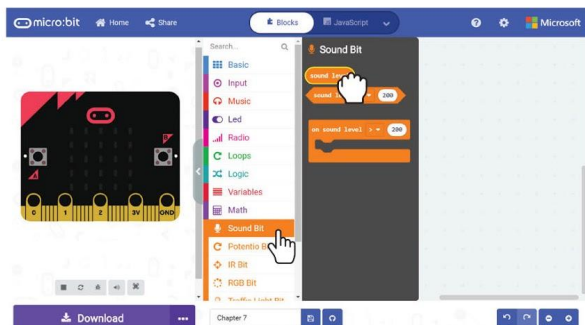


Gyerünk Kódolni!

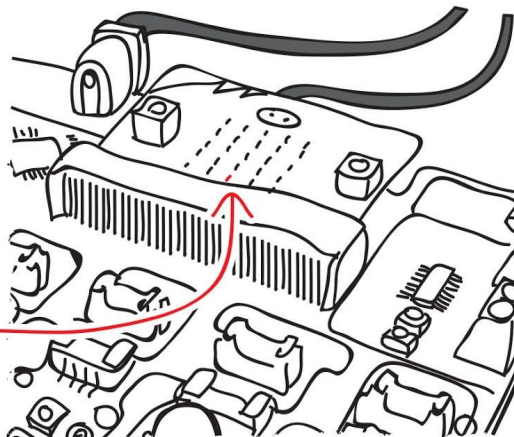
1.lépés Hozzon létre egy új projektet a MakeCode Editor-ban, és adja hozzá az EDU:BIT kiterjesztést (lásd a 40. oldalt). Kattintson a [Led] kategóriára és válassza a [plot bar graph of_up to_] blokkot. Helyezze a kijelölt blokkot a [forever] blokkba.



2.lépés Kattintson a [Sound Bit] kategóriára és válassza a [sound level] blokkot. Helyezze a blokkot a [plot bar graph of_up to_] blokkba és változtassa meg a második értéket 0-ról 1023-ra.



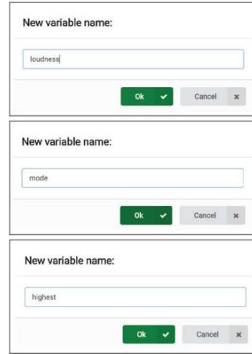
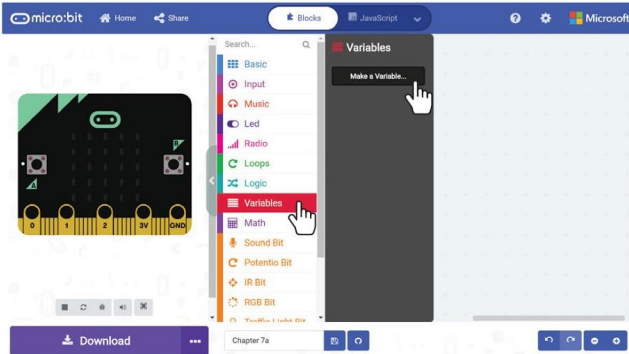
3.lépés Töltse fel a kódot az EDU:BIT-re. Figyelje meg a LED-mátrix kijelzőjét, miközben tapsol, vagy dobol az ujjával az asztalon.



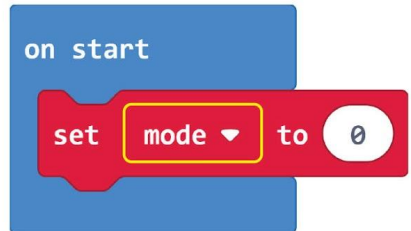
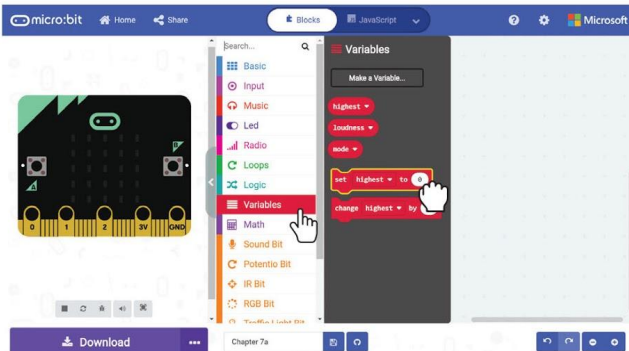


Most fordítsuk az EDU:BIT-et taps-o-méterré.

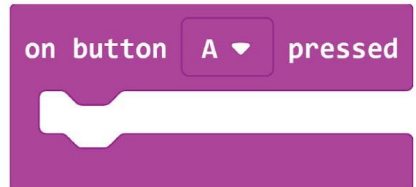
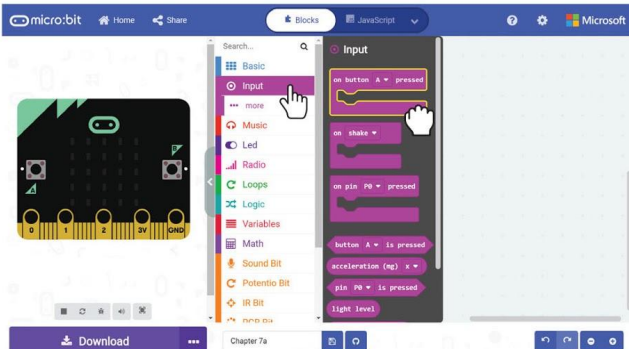
4.lépés Kezdjünk egy új projektet és adjuk hozzá az EDU:BIT kiterjesztéseit. Kattintson a **[Variables]** kategóriára és utána a **[Make a Variable]**. Írja be a 'mode' elemet a felbukkanó ablakba, majd kattintson az OK gombra. Hozzon létre még két változót, amelyeknek a nevei 'loudness' és 'highest'.



5.lépés Vegyen ki egy **[set_to_]** blokkot a **[Variable]** kategóriából. Helyezze be a blokkokat az **[on start]** nyílásába. Állítsa a változót 'mode'-ra.

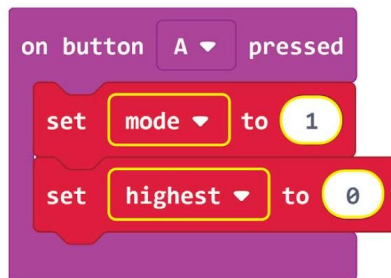
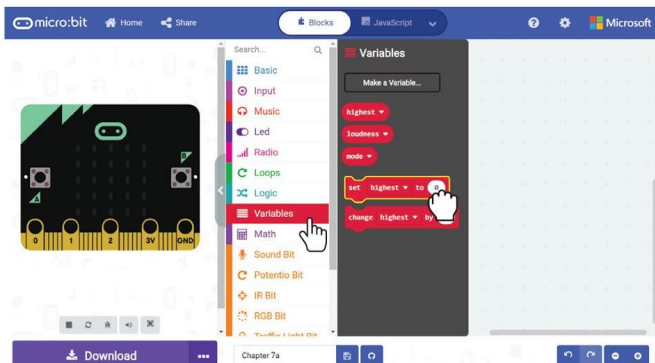


6.lépés Kattintson **[Input]** kategóriára és válassza az **[on button_pressed]** blokkot.

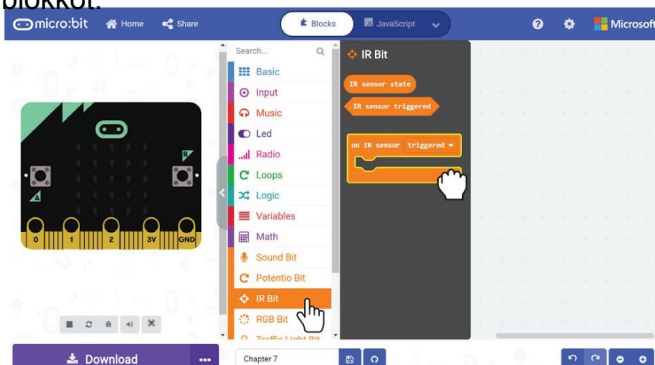




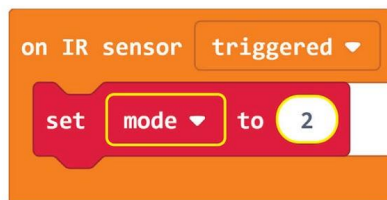
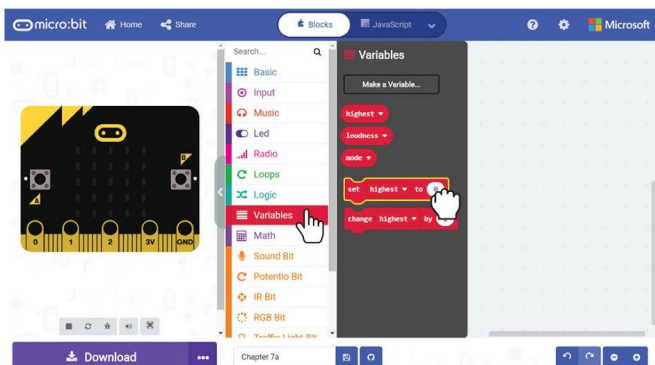
7.lépés Kell két `[set_to_]` blokk a `[Variables]` kategóriából és tegye az `[on button A pressed]` nyílásába. Az első `'mode'` változónak állítsa át az értékét 1-re, a második változónak `'highest'` az alapértelmezett 0 érték maradjon.



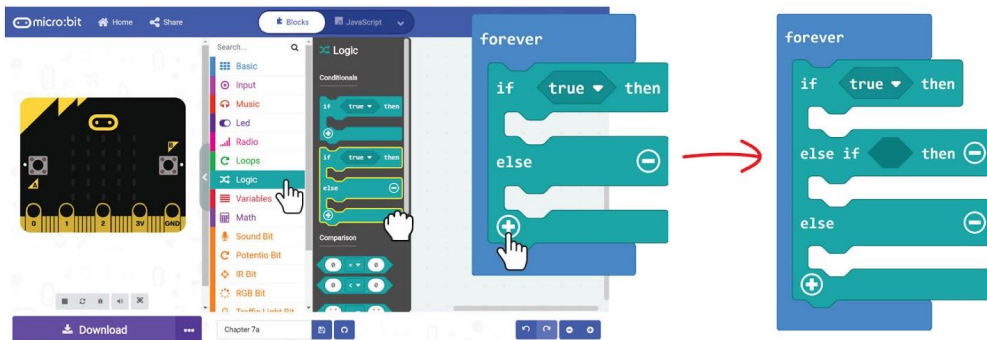
8.lépés Kattintson az `[IR Bit]` kategóriára és válassza az `[on IR sensor triggered]` blokkot.



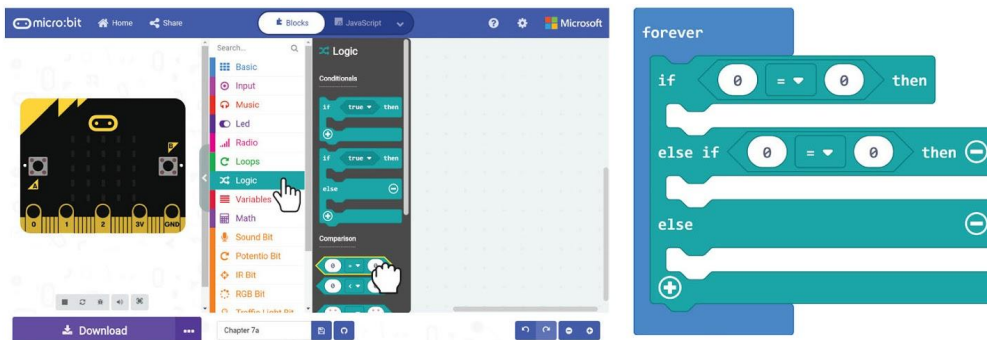
9.lépés Kattintson a `[Variables]` kategóriára, majd válassza a `[set_to_]` blokk lehetőséget. Csatlakoztassa a blokkot az `[on IR sensor triggered]` nyílásba, és állítsa a változót `'mode'`-ra, az értéket pedig 2-re.



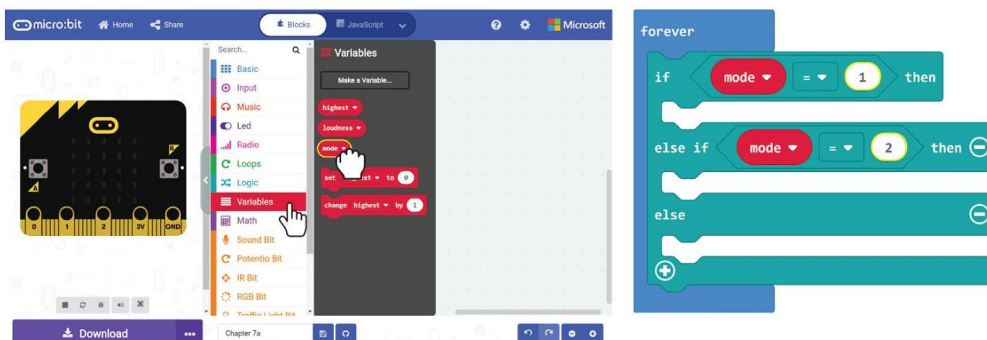
10.lépés Kattintson a [Logic] kategóriára és válassza az [if-then-else] blokkot. Csúsztassa a blokkot a [forever] nyílásba. Kattintson a plusz ikonra, else-if feltételt szeretne hozzáadni a blokkhoz.



11.lépés Kattintson a [Logic] kategóriára és válassza a [=] összehasonlító blokkot. Másold le a blokkokat, és kattints a blokkokra az [if-then-else] blokk feltételeire.

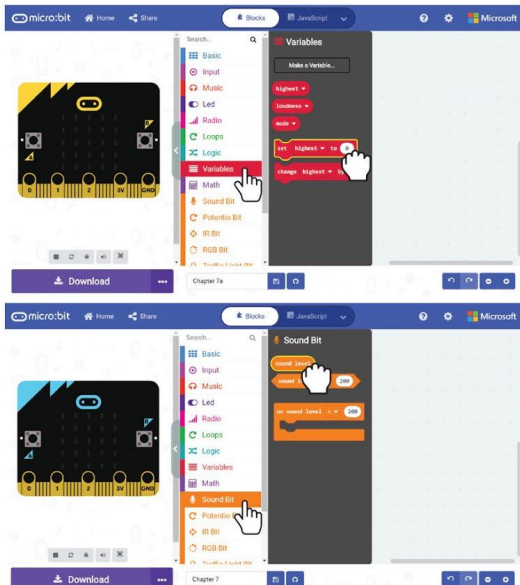


12.lépés Válassza a [mode] elemet a [Variable] kategóriából, és tegye az összehasonlító blokkok bal nyílásába. Állítsa a többi feltételt 1-re, illetve 2-re.





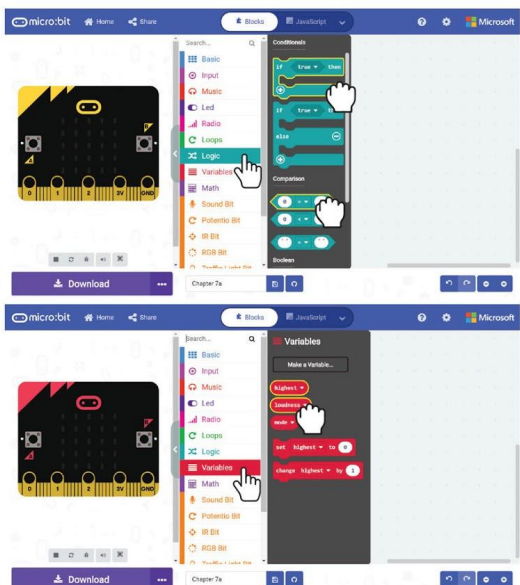
13.lépés Válassza ki a [set_to_] blokkot a [Variables] kategóriából, és kattints a blokkra az első [if-then-else] nyíláshoz. Állítsa a változót 'loudness'-re, és kattintson a [sound level] blokkot a [Sound Bit] kategóriáról az értékhelyre.



```

forever
  if mode = 1 then
    set loudness to sound level
  else if mode = 2 then
  else
  
```

14.lépés Kattintson a [Logic] kategóriára, majd válassza az [if-then] blokkot és az [=] összehasonlító blokkot. Változtassa a = szimbólumot a > szimbólumra. Helyezzük a [loudness] és a [highest] a [Variables] kategóriából az összehasonlító blokk nyílásaiba.

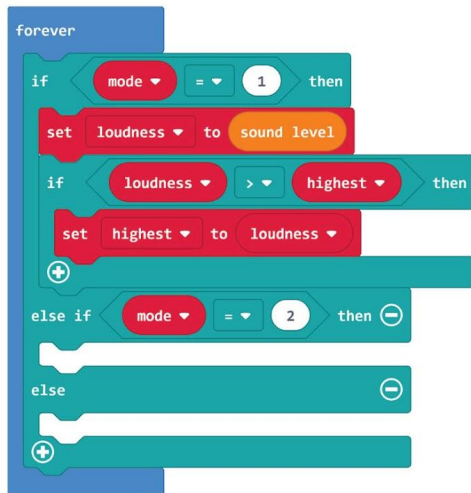
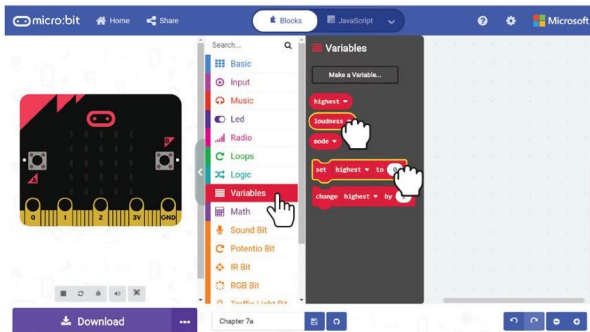


```

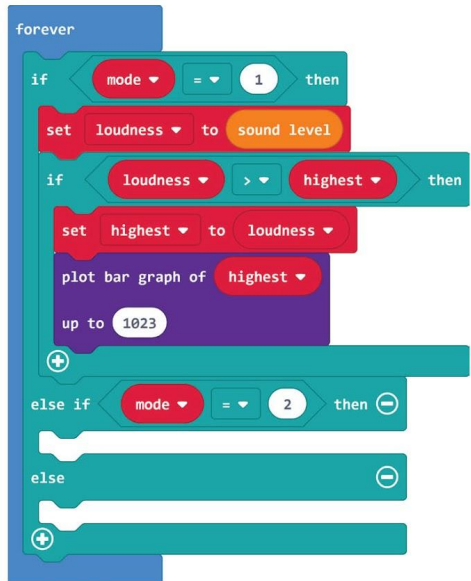
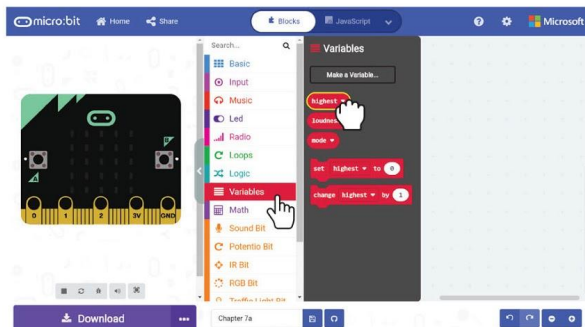
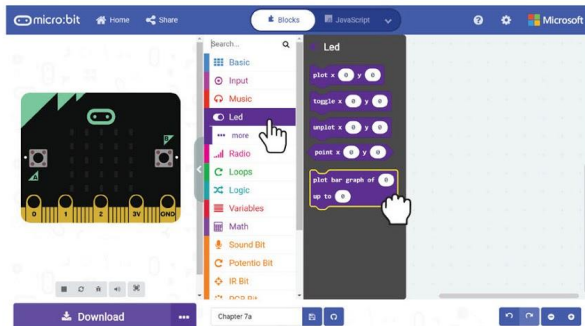
forever
  if mode = 1 then
    set loudness to sound level
    if loudness > highest then
  else if mode = 2 then
  else
  
```

7. fejezet : Sound Bit

15.lépés Kattintson a [Variables] kategóriára, majd válassza a [set_to_] blokk lehetőségét. Helyezze a blokkot az [if-then] részbe és a [loudness] a [Variables] kategóriából az értékhelybe.

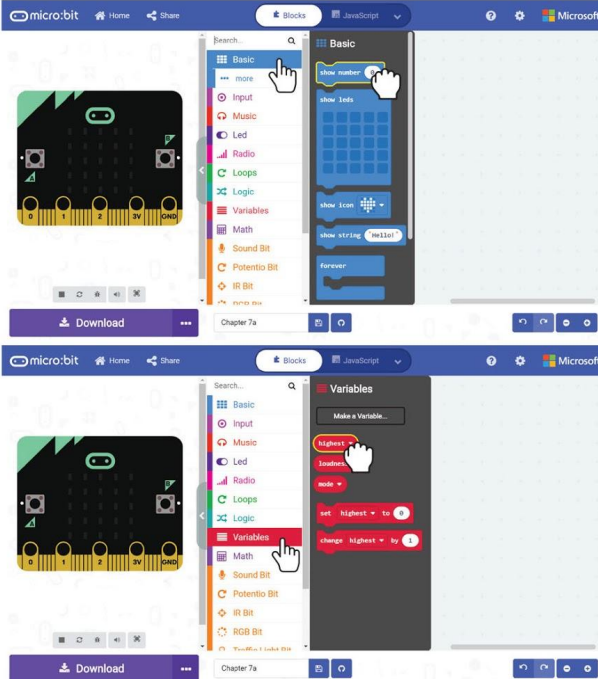


16.lépés Kattintson a [Led] kategória elemre, és válassza a [plot_graph_of_up to_] blokkot. Kattintson a [Variables] kategóriára, és válassza a [hangosság] blokk lehetőségét. Helyezze a blokkot a [plot_gram of_up to_] blokkra, és változtassa meg az értéket 1023-ra.





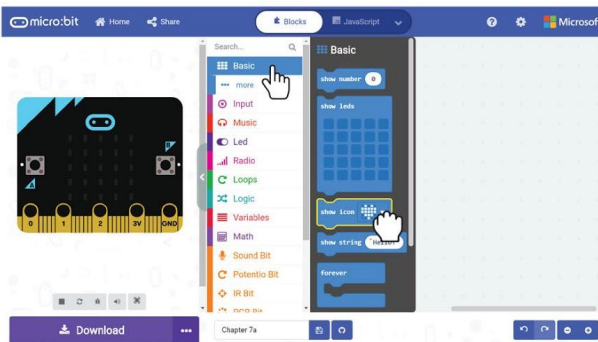
17.lépés Kattintson a [Basic] kategóriára és válassza a [show number] blokkot. Helyezze az [if-then-else] blokk második nyílásához. Szerezze meg a [highest] blokkot a [Variable] kategóriából, és illessze a [show number] blokk érték helyére.



```

forever
  if mode = 1 then
    set loudness to sound level
    if loudness > highest then
      set highest to loudness
      plot bar graph of highest
      up to 1023
    else if mode = 2 then
      show number highest
    else
  
```

18.lépés Kattintson a [Basic] kategóriára és válassza a [show icon] blokkot. Helyezze az utolsó nyílásba az [if-then-else] blokkot.



```

forever
  if mode = 1 then
    set loudness to sound level
    if loudness > highest then
      set highest to loudness
      plot bar graph of highest
      up to 1023
    else if mode = 2 then
      show number highest
    else
      show icon
  
```

Itt a teljes kód:

Indításkor állítsa az üzemmódot 0-ra.

```
on start
  set mode to 0
```

Az A gomb megnyomásakor állítsa az üzemmódot 1-re, és állítsa a "legmagasabb" változót 0-ra

```
on button A pressed
  set mode to 1
  set highest to 0
```

Az IR bit aktiválásakor változzon át a mód 2-re.

```
on IR sensor triggered
  set mode to 2
```

Mindig ellenőrizze az aktuális módot.

```
forever
  if mode = 1 then
    set loudness to sound level
    if loudness > highest then
      set highest to loudness
      plot bar graph of highest
      up to 1023
    else if mode = 2 then
      show number highest
    else
      show icon [LED Matrix]
```

Ha mód = 1 (azaz az A gombot megnyomják), ábrázolja a legmagasabb észlelt zajszint oszlopdiagramját a LED mátrix kijelző világításával.

Minél nagyobb a zaj, annál több LED világít; és fordítva.

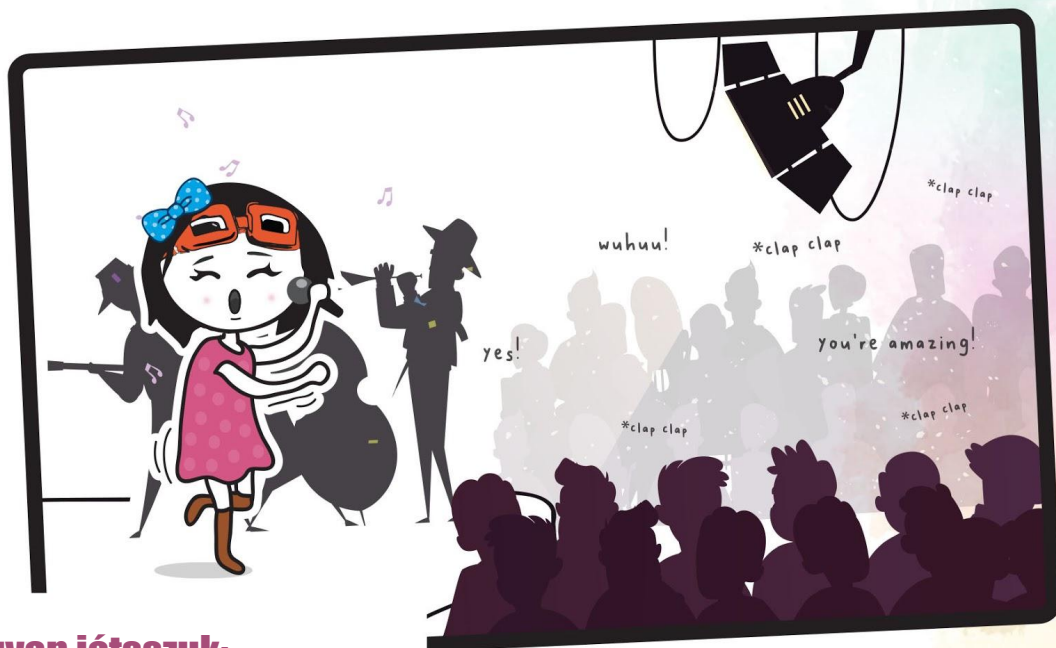
Ha mód = 2 (azaz IR szenzort vált ki), akkor mutassa meg a "legmagasabb" változó aktuális értékét (észlelt zajszint).

Ha a mód nem 1 vagy 2, akkor mutassa meg a szív ikont.



19.lépés Töltse fel a kódot az EDU: BIT-re, és akkor készen áll egy tapsmérővel a Tehetség Idő Show-jához.

Játszunk!



Hogyan játszunk:

- A "versenyzők" kapnak időt egy rövid előadás előkészítésére, külön-külön, párban vagy csapatban. Választhatnak éneklés, táncolás, vagy akár vicc mondás között.
- Amikor mindenki készen áll, felváltva mutassák be tehetségüket. Minden előadás után a "közönség" tapsal tapsol - minél jobban élvezik a műsort, annál hangosabban tapsolnak.
- Miután a taps elhallgatott, indítsa el az IR bitet a pontszám (a legmagasabb rögzített hangszint) görgetéséhez.
- A nyertes az az egyén, pár vagy csapat, amely a lehangosabb tapsot kapja. Jó szórakozást!

Értelmezzük a program működését

Ha egy programban több feladatunk van, akkor eseményindítókkal válthatunk egyik feladattól a másikig. A program zökkenőmentes futtatásához egy örök hurkot használunk az aktuális mód állandó ellenőrzésére, majd a megfelelő kódblokk(ok) végrehajtására.

Indításkor állítsa az üzemmódot 0-ra (azaz készenléti üzemmódra).

```
on start
  set mode to 0

on button A pressed
  set mode to 1
  set highest to 0

on IR sensor triggered
  set mode to 2
```

Mindig ellenőrizze az aktuális módot, majd futtassa az adott módhoz rendelt feladatot (kódblokkok).

```
forever
  if mode = 1 then
    set loudness to sound level
    if loudness > highest then
      set highest to loudness
      plot bar graph of highest
      up to 1023
    else if mode = 2 then
      show number highest
    else
      show icon [grid icon]
```

Az eseményindító blokkok az egyik üzemmódból a másikba váltáshoz futás közben.



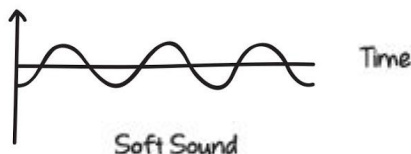
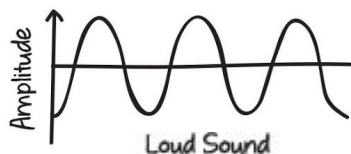
Ha további módokat / feladatokat kell felvenned, mindig több eseményindító blokkot, például [rázáskor] és [hangszint> _], használhatsz a kódodhoz, és kattints a (+) gombra az [if-then-else] blokk további feltételek hozzáadásához.

Vicces Tény!



A hang akkor keletkezik, amikor egy tárgy rezeg, például egy dob, ha megütik. A rezgés hatására a körülötte lévő légmolekulák (közeg) rezegnek és hanghullámokat hoznak létre.

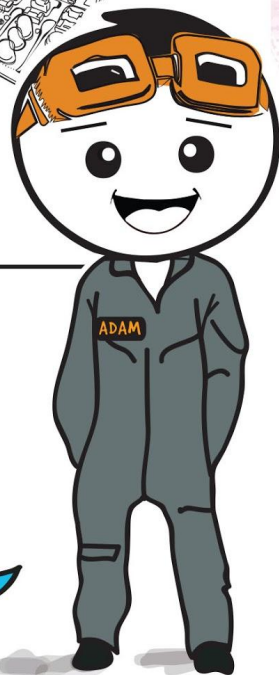
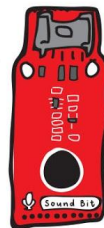
A **hangérzékelő** egy olyan modul, amely érzékeli a hanghullámok intenzitását (vagyis a hangosságot) és alakítja át ezt villamos jellé.



Bizonyos szempontból a hangérzékelő hasonlóan működik, mint a fülünk, amely a levegőben lévő rezgéseket elektrokémiai jelekké alakítja, amelyeket az agy által ismert hangokká alakít.



Sound Detected



Íme néhány gyakori alkalmazás:

- Betörésjelző zajérzékelővel
- Hangvezérlésű világítás
- Baba monitor

Gondolod, hogy a hangérzékelőnk képes bármilyen zajt észlelni a világűrben? Miért vagy miért nem?



Alkalmazás kihívás

Program EDU: BIT, hogy osztálytermi zajmonitorként működjön. A zajszint jelzésére világítson a LED-ekkel a Traffic Light Bit-en.

Zajszint	Zajszint tartomány	Közlekedési lámpa bit
Túl hangos; kérem tekerje le a hangerőt.	() 1023-ig	Piros LED
Kissé zajos; kérem ne felejtse a hangerőt szabályozni	() között ()	Sárga LED
Elfogadható zajszint. Jó!	0 és () között	Zöld LED

Íme néhány tipp az Ön számára ...

1. tipp: Előzetesen meg kell határoznia az egyes zajszintek küszöbértékeit.
2. tipp: A stabilabb figyelés érdekében rendszeres időközönként kapja meg a zajszint átlagos értékét.



Túl könnyű? Próbáljon eggyel feljebb jutni.
Módosítsa a kódot úgy, hogy a küszöbértékek viszonyuljanak a potenciométer értékéhez.



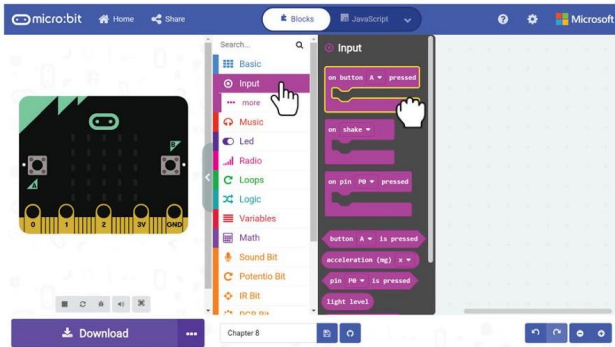
Gyerünk pörgetni és menjünk körbe-körbe
DC motor



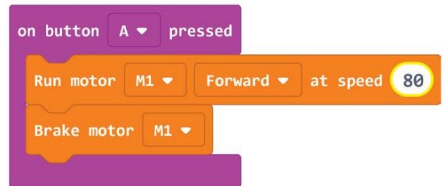
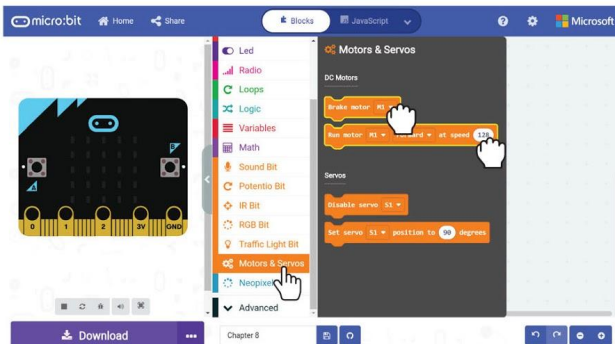
8. fejezet : DC motor

Gyerünk kódolni!

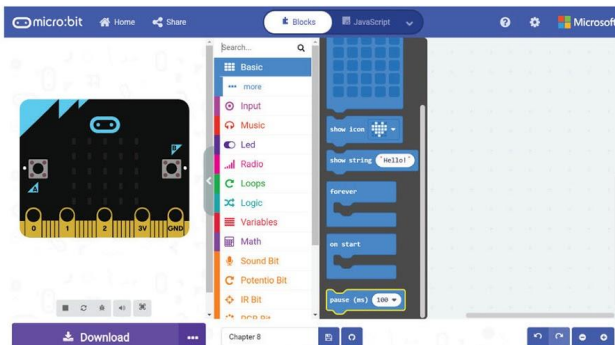
1.lépés Indítson el egy új projektet a MakeCode Editorban, és adja az EDU: BIT kiterjesztést (lásd: 40. oldal). Kattintson az [Input] kategória elemre, majd válassza a [on button_pressed] blokkot.



2.lépés Kattintson a [Motors&Servos] kategóriára és adja a [Run motor__at speed_] blokkot és a [Brake motor_] blokkot a kódoláshoz. Változtassa a sebességet 80-ra.

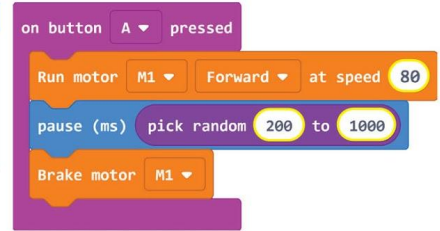


3.lépés Kattintson a [Basic] kategóriára és válassza a [pause_] blokkot. Helyezze ezt a blokkot a [Run motor__at speed_] és a [Brake motor] blokk közé.

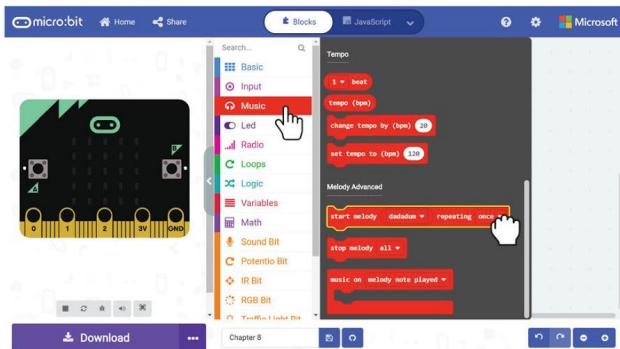




4. lépés Kattintson a [Math] kategóriára és válassza a [pick random_to_] blokkot. Helyezze a blokkot a [pause_] blokkba, majd módosítsa az értékeket 200, illetve 1000-re.



5. lépés Kattintson a [Music] és válassza a [start melody_repeating_] blokkot. Változtassa a dallamot "ba ding" -re (vagy válasszon tetszőleges dallamot).

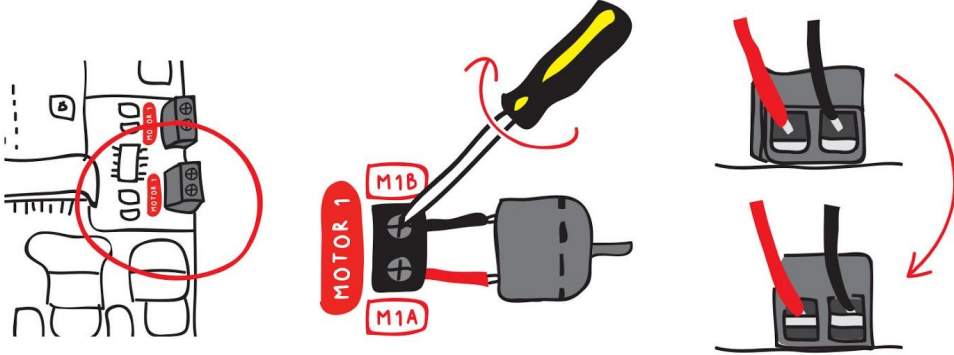


6. lépés Töltse fel a teljes kódot az EDU:BIT-re.

Használhatjuk ezt a kódot minden olyan játékhoz, amely véletlenszerű forgást igényel. Alapvetően a motor forogni kezd, amikor beindul, majd véletlenszerű idő után leáll.

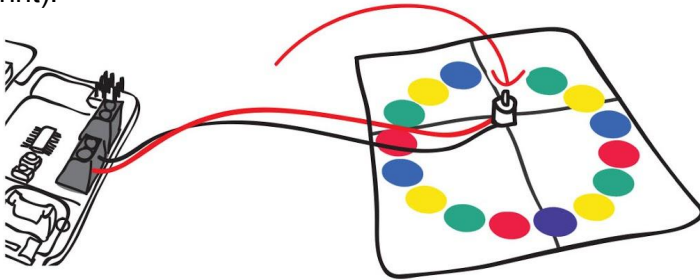


7.lépés Csatlakoztassa az egyenáramú motort a MOTOR 1 csatlakozóhoz - (i) helyezze be a szabadon lévő vezetéket, majd (ii) húzza meg a csavart a mellékelt csavarhúzóval a csatlakozás biztosításához, és tartsa a helyén.

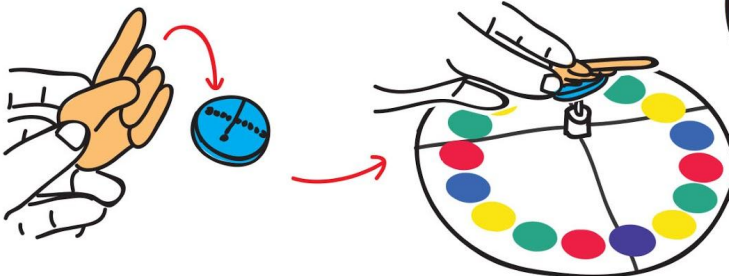


A teszteléshez nyomja meg a sárga gombot (A gomb). Ha a motor nem forog, ellenőrizze, hogy a vezeték csatlakozása biztonságos-e a terminálon, és az EDU: BIT be van-e kapcsolva.

8.lépés Használjon extra ragasztót, például kétoldalas szalagot vagy forró ragasztót, hogy az egyenáramú motort rögzítse a Twister kerék középhez (az ábra szerint).



9.lépés Húzza ki a mutatót, és ragasztóval rögzítse a műanyag koronghoz. Ezután rögzítse a tárcsát a motor tengelyére.



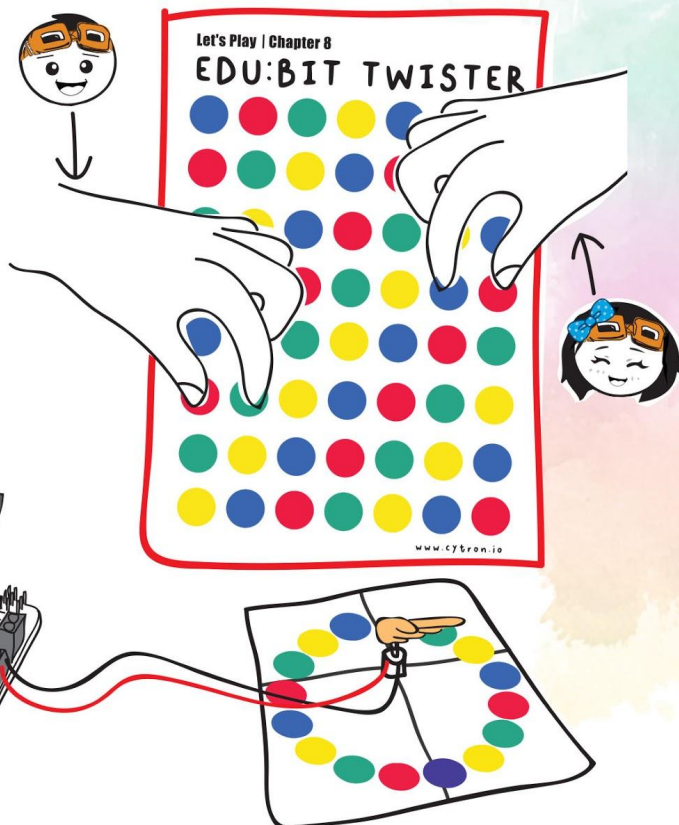
A dobozban Twister kerék, mutató és játéktérkép található.



Gyerünk játszani! Gyerünk forgass!

Játék beállítása

- Fektesse le a játéktérképet az asztalra. Két játékos ül egymással szemben. Ha vannak további játékosok, akkor osszuk el a közöttük lévő távolságot (max 4 játékos).
- A játékvezető a közelben ül, és az EDU: BIT twister kerék könnyen elérhető.



Hogyan játszuk:

Ebben a játékban a játékosok felváltva helyezik ujjukat a játéktérkép színes körökre a játékvezető utasítása szerint.

A játékvezető feladata, hogy megnyomja a sárga gombot (vagy az A gombot), hogy megpördítse a mutatót, majd felmutatja az ujját és a színt, amelyre a mutató mutat, például: "mutatóujj; piros".

Amikor rád kerül a sor, meg kell hallgatnod a játékvezető utasítását, és a behívott ujjat a megfelelő színű körre kell helyezni, és meg kell próbálnod áthelyezni egy másik azonos színű körbe.

Ha nem sikerül sikeresen végrehajtania egy lépést, akkor kiesik a játékból.

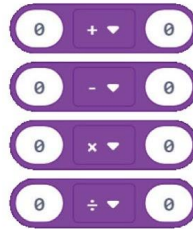
A játékban utolsó játékos maradt a **GYŐZTES!**



További blokkok

A [Maths/Matematika] kategória blokkjaival számtani műveleteket hajthat végre a változókon.

#1 A következő blokkok segítségével összeadhatja, kivonhatja, szorozhatja vagy eloszthatja.

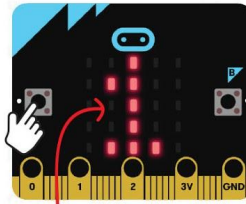


#2 A [remainder of $_ \div _$] blokk segítségével megtudhatja, mennyi van hátra, ha az egyik szám nem oszlik el egyenletesen a másik számra.

For example:

```
on button A pressed
  show number remainder of 13 ÷ 2
```

$$\begin{array}{r} 6 \\ 2 \overline{)13} \\ \underline{12} \\ 1 \end{array}$$



remainder

#3 Használhatja a [remainder of $_ \div _$] blokkot annak eldöntésére, hogy egy szám páratlan vagy páros-e. Ha a maradék értéke "1", akkor az "páratlan"; ha a maradék értéke "0", akkor az "páros". Próbálja ki!

```
on start
  set Counter to 0

on button A pressed
  change Counter by 1
  show number Counter
  if remainder of Counter ÷ 2 = 0 then
    show string "even"
  else
    show string "odd"
```



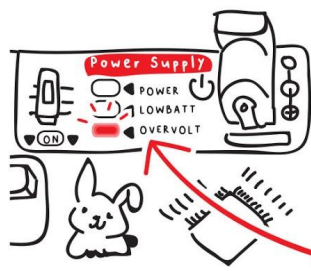
Észreveszi, hogy az EDU: BIT görgetve "páratlan", amikor először megnyomja az A gombot (1 = páratlan szám), majd a következő gomb megnyomásakor "páros" (2 = páros szám)?
Mi jelenik meg, amikor az A gombot ??

Figyelem!



Az egyenáramú motor, vagy közismertebben **DC motor**, egy forgó elektromos eszköz, amely az elektromos energiát mechanikai energiává alakítja.

Bemeneti feszültséget kell alkalmaznunk, hogy az egyenáramú motor forogjon. A centrifugálási sebességet a bemeneti feszültség beállításával szabályozhatjuk. Minél nagyobb a bemeneti feszültség, annál gyorsabban forog a motor. Az egyenáramú motor ajánlott feszültsége az EDU:BIT-ben 3,6 V-6 V.



Figyelem!

Nagyfeszültség alkalmazása egy motorra (meghaladja az ajánlott feszültséget) hosszú távon lerövidíti a motor élettartamát.



Könnyedén szabályozhatja az egyenáramú motor forgási irányát és sebességét a következő programozási blokkal.

Run motor M1 Forward at speed 255

✓ M1 M2

✓ Forward Backward

Speed 255

Forgó irány

Két DC motor terminál található az EDU:Bit-en, válassza ki a megfelelőt.

Ez a relatív érték 0 és 255 között mozog. Minél nagyobb az érték, annál gyorsabban forog a motor.

Tudta, hogy az EDU: BIT beépített motor teszt áramkörrel rendelkezik?

Nyomja meg a fehér gombokat (M1A, M1B, M2A és M2B jelöléssel), hogy ellenőrizze, hogy a csatlakozása biztonságos-e és a motor jól működik-e.



Alkalmazás kihívás

EDU: BIT a funkcióként hangvezérelt ventilátor, amelynek sebességét a Potentio Bit vezérli.

On start	Szív ikon (vagy bármelyik tetszőleges ikon) megjelenítése Állítsa a változó módot 0 értékre
On sound level > _ (küszöbérték)	Változtassa meg a változó módot 1-gyel
Forever	Állítsa be a „Speed” változót a potenciométer értékének feltérképezéséhez az alacsony 0 magas 1023 és az alacsony 0 magas 255 érték között. Mindig ellenőrizze az üzemmódot Ha a mód páros szám, akkor az M1 fékezze a motort. Egyébként, ha a Mode páratlan szám, akkor az M1 motort a „Speed” változó értékének megfelelően futtassa (a potenciométer leolvasása alapján feltérképezve).



Itt van néhány tipp neked:

Tipp #1 A motor aktiválásához és leállításához meg kell határoznia a zajszintet (azaz küszöbértéket).

Tipp #2 Két változót kell létrehoznia: Mode és Speed.

Tipp #3 Csatlakoztassa a ventilátor lapátjait a motor tengelyéhez, majd futtassa a programot. Ha nem érzi, hogy levegő fúj, amikor a motor forog, akkor meg kell változtatnia a centrifugálást, meg kell változtatnia a forgási irányt a kódjában.

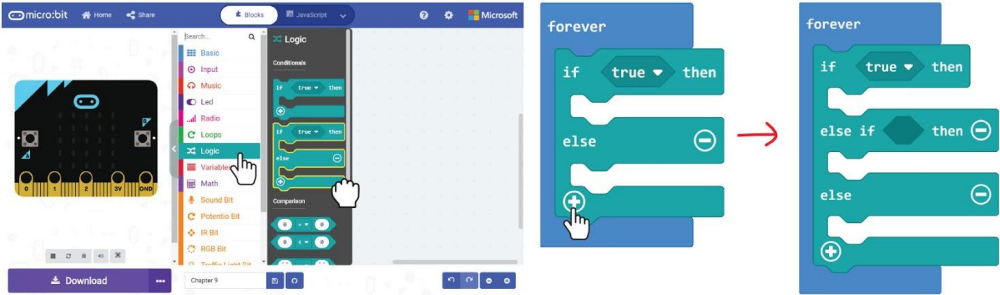
Büntető lövés....GÓL!!!
Servo motor

Ready..
Get set..
GO!!!!!!

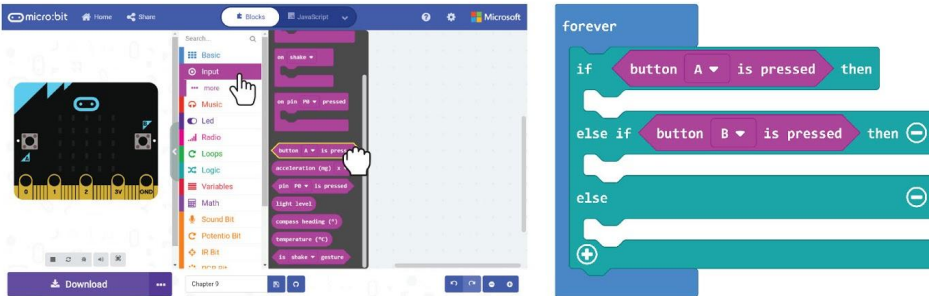


Györünk kódozni!

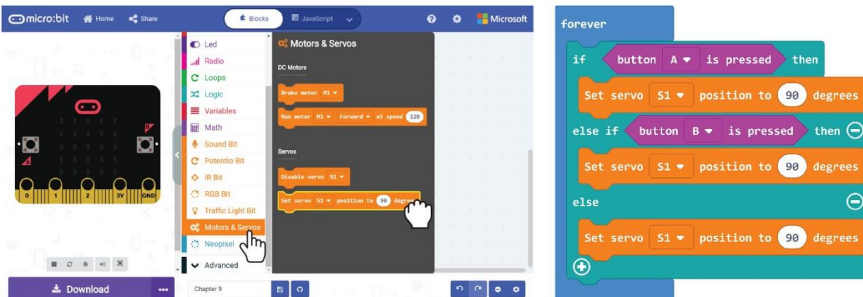
1.lépés A MakeCode Editorban, hozzon létre egy új projectet az EDU:BIT kiterjesztésekkel (hivatkozás a 40. oldalon). Kattintson a **[Logic]** kategóriára és válassza a **[if-then-else]** blokkot. Helyezze a blokkot a **[forever]** nyílásba. Kattintson a plussz ikonra és adjon hozzá további else-if feltételt a blokkhoz



2.lépés Kattintson az **[Input]** kategóriára és válassza a **[button is pressed]** blokkot. Másolja le a blokkot és kattintson az **[if-then-else]** blokk feltételeire. Változtassa a második blokkot 'B' gombra.



3.lépés Kattintson a **[Motors & Servos]** kategóriára és válassza a **[Set servo_position to_degrees]** blokkot. Másolja a blokkot és csatolja az **[if-then-else]** blokkhoz.

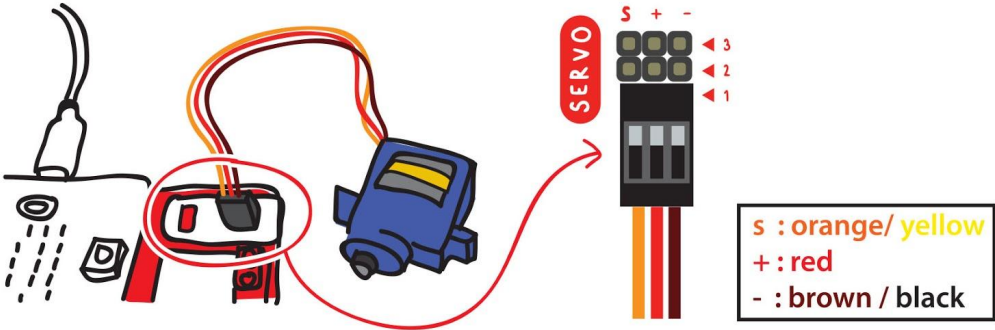




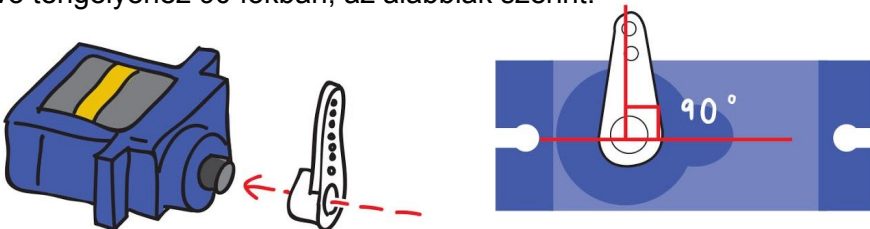
4.lépés Módosítsa az első és a második blokk értékét **30**-ra, illetve **150**-re. Töltse a kódot EDU:BIT-re.

```
forever
  if button A is pressed then
    Set servo S1 position to 30 degrees
  else if button B is pressed then
    Set servo S1 position to 150 degrees
  else
    Set servo S1 position to 90 degrees
```

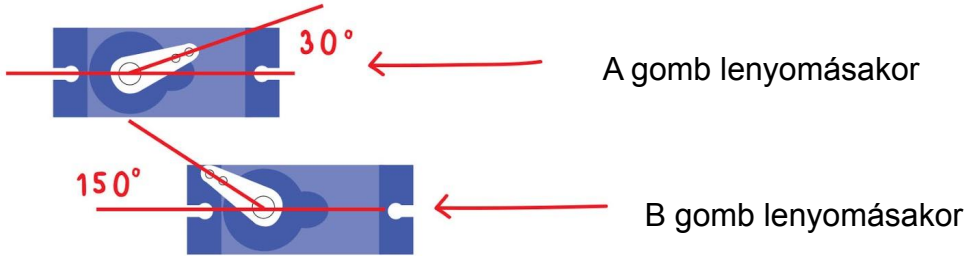
5.lépés Dugja be a szervomotor kábelét az EDU: BIT 1. servo portjába az alábbiak szerint.



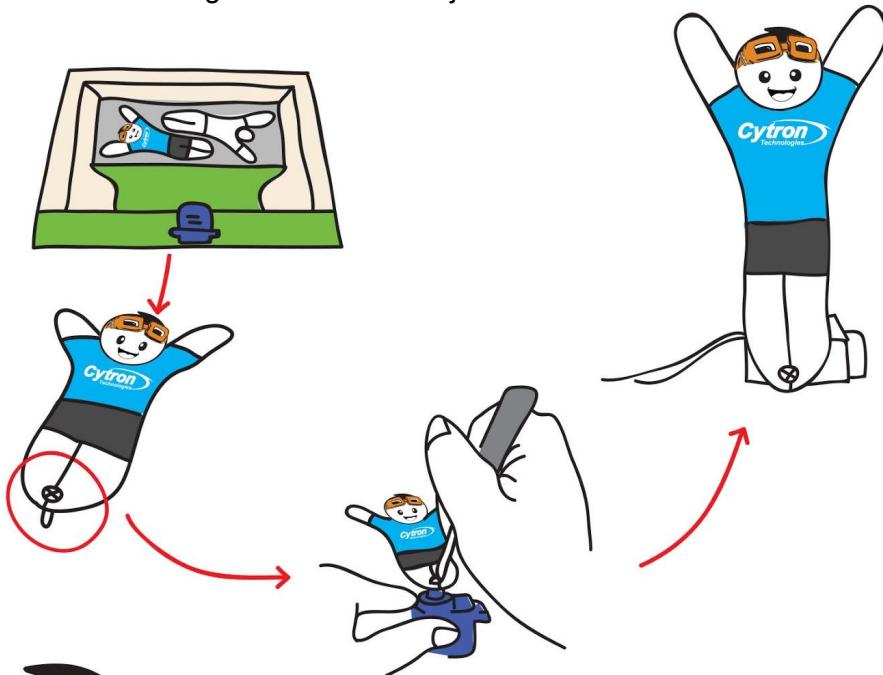
6.lépés Kapcsolja be az EDU:BIT-et, majd csatlakoztassa a szervokart a motor servo tengelyéhez 90 fokban, az alábbiak szerint.



7.lépés Nyomja meg az A gombot, majd a B gombot a teszthez.



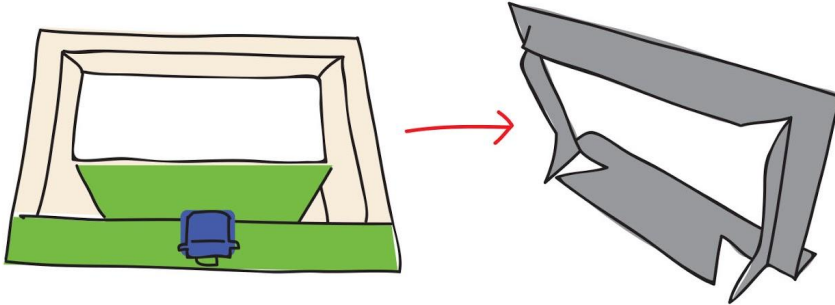
8.lépés Vegye ki a kapust a rendelkezésre álló forráskártyáról. A mellékelt csavarhúzóval és csavarral rögzítse a kapust a servo karjához. Használjon extra ragasztót, például kétoldalas szalagot vagy forró ragasztót, hogy a kapust szilárdan rögzítse a servo karjához.



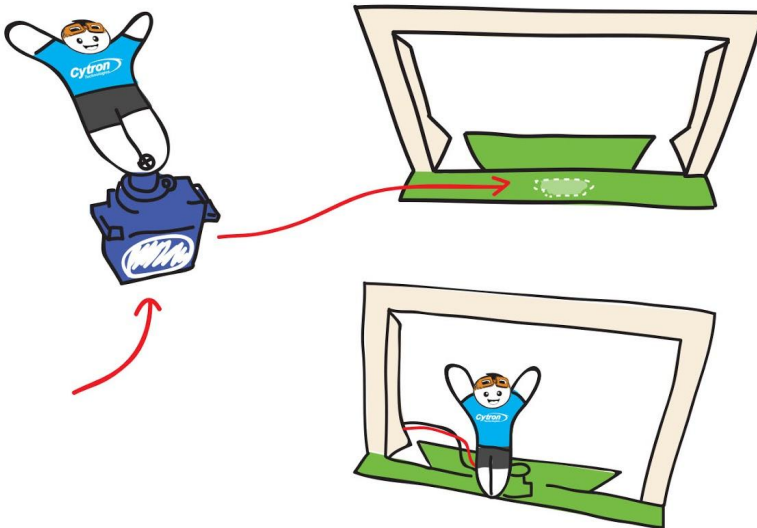
Ha úgy tetszik, használhatja a rendelkezésre álló másik előugró alakot, és megtervezheti saját mezét a kapusnak.



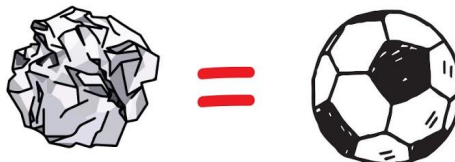
9.lépés Nyissa ki a kapufát, és állítsa be az alábbiak szerint.



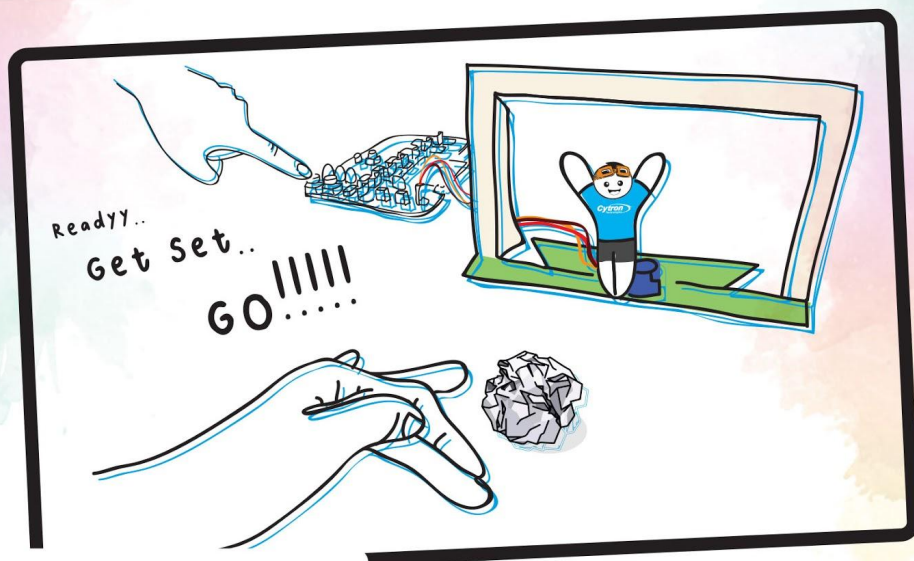
10.lépés Használjon erős ragasztót, például kétoldalas szalagot vagy forró ragasztót a szervomotor szilárd rögzítéséhez a jelzett módon.



11.lépés Gyúrjon össze egy kis papírt, hogy helyettesítő "focilabda" legyen, és mindannyian egy szórakoztató büntetőlövésre készülünk. Kész vagy?



Kezdődjön a játék!



Hogyan játszuk:

Állítsa be a kapufát, és jelölje ki a tizenegyest (kb. 1 méterre a kaputól, állítsa be a távolságot a fiatalabb játékosok számára).

A játékosok felváltva Lövők és Kapusok.

A Lövő a kapu felé lövi a labdát.

A kapus megpróbálja blokkolni a labdát úgy, hogy balra mozog a sárga gomb (A gomb) megnyomásával, vagy jobbra a kék gomb (B gomb) megnyomásával.

Egy fordulóban minden játékos 5 lehetőséget kap. Az a játékos nyer, aki a legtöbb gólt szerzi.



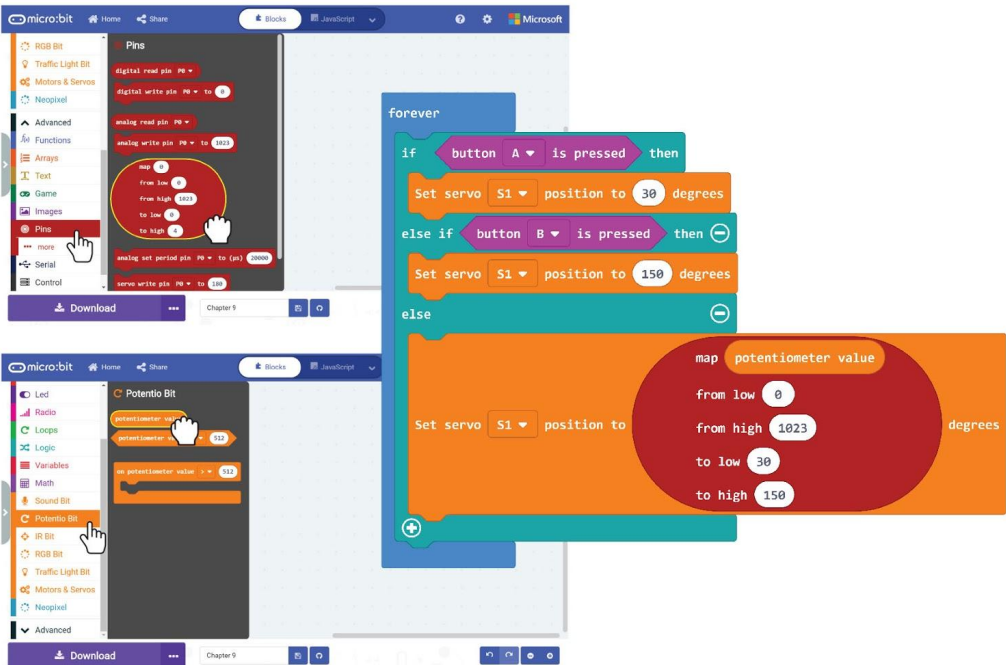
Tudtad? Tizenegyesekkel játsszák le a győztes csapat meghatározását egy futball meccsen, amikor a rendes játék végén egyenlő az eredmény, és a hosszabbítás után is döntetlen marad. Minden csapat öt lövést kap. A játékos a tizenegyesből a kapu felé lö, amelyet csak az ellenfél kapusa őriz. A győzelem annak a csapatnak jár, amelyik több gólt szerez.



Korábbi kódunkban a kapust balra vagy jobbra mozgathatjuk a nyomógombok megnyomásával. Módosíthatjuk a kódot a Kapus helyzetének ellenőrzésére a Potentio Bit segítségével.

12.lépés Kattintson az **[Advanced]** kategóriára és válassza a **[Pins]** kategóriát. Adja hozzá a **[map_from low_from high_to low_to high_]** blokkot a kódolásához.

13.lépés Kattintson a **[Potentio Bit]** kategóriára és válassza **[potentiometer value]** blokkot. Kapcsolja a blokkhoz hozzá a **[map_from low_from high_to low_to high_]** blokkot és változtassa meg az utolsó két számot 30-ra és 150-re.



14.lépés Töltse fel kódját az EDU:BIT-re. Mostantól a Potentio Bit segítségével irányíthatja a kapust. Jó szórakozást!

Ha önállóan szeretné gyakorolni a büntetőrúgások végrehajtását, módosíthatja a kódot "gyakorló módra" úgy, hogy a kapust folyamatosan balra és jobbra lengeti. Próbálja ki!



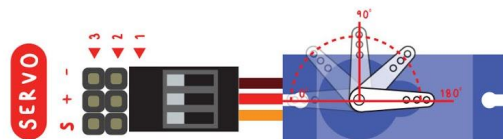
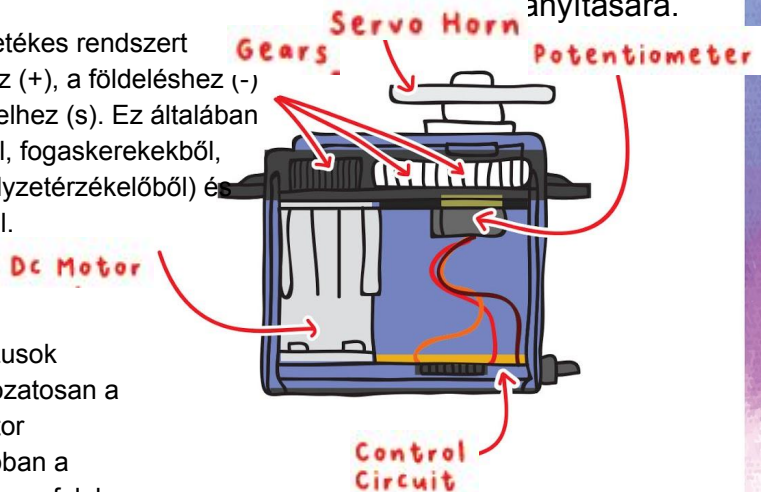
Figyelem!



Az EDU: BIT készlet szervomotorja RC (rádió vezérlés) szervóként is ismert. Széles körben használják RC játkéjrművekben és kis robotokban mozgásuk anyítására.

A szervomotor háromvezetékes rendszert használ az áramellátáshoz (+), a földeléshez (-) és a vezérléshez vagy a jelhez (s). Ez általában egy egyenáramú motorból, fogaskerekekből, egy potenciométerből (helyzetérzékelőből) és egy vezérlő áramkörből áll.

A beépített vezérlő impulzusok formájában fordítja le fokozatosan a parancsokat. A szervomotor folyamatosan forog, és abban a helyzetben marad, amely megfelel a kapott impulzusoknak.



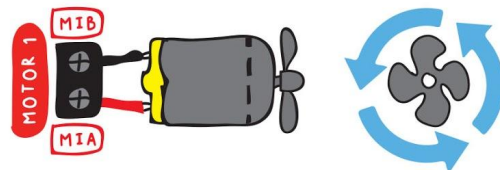
0-180° rotation

Servo Motor

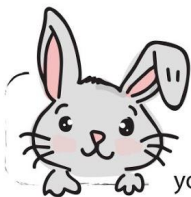
VS

DC Motor

360° rotation



A folyamatosan forgó egyenáramú motorral ellentétben a szervomotor forgatását a kívánt szögig tudjuk szabályozni 0 és 180 fok közötti tartományban.



youtu.be/okxooamdAP4

Alkalmazás kihívás

Az EDU: BIT program tud metronómként funkcionálni. Bekapcsoláskor a mutatót (a szervomotorhoz rögzítve) állandó ütemben a balra, majd ismételten jobbra lendítse. Állítsa be a tempót a Potentio Bit vezérléséhez. A sárga gomb (A gomb) megnyomásakor mutassa meg az aktuális tempót (pl. 120 ütés / perc).



Itt van néhány tipp...

Tipp #1 Két változót kell létrehoznia: Tempo és Delay

Tipp #2 A tipikus metronóm tartomány 40-200 ütés / perc.

Tipp #3 A 60 ütés / perc (vagy percenkénti ütem) tempó azt jelenti, hogy a mutató percenként összesen 60-szor, azaz másodpercenként egyszer mozog az egyik végétől a másikig.

Tipp #4 Minél gyorsabb a tempó, annál kisebb a késés.



„A metronóm egy olyan eszköz, amely hallható kattánást vagy más hangot produkál rendszeres időközönként, amelyet a felhasználó beállíthat, általában ütemenként percenként (bpm). A zenészek a készülék segítségével gyakorolják a rendszeres pulzusra való játékot. A metronóмок általában szinkronizált vizuális mozgást tartalmaznak.”

-wikipedia-

Mesterelme, Fel tudja törni a titkos kódot?
RGB Bit

Code Breaker



Code Maker





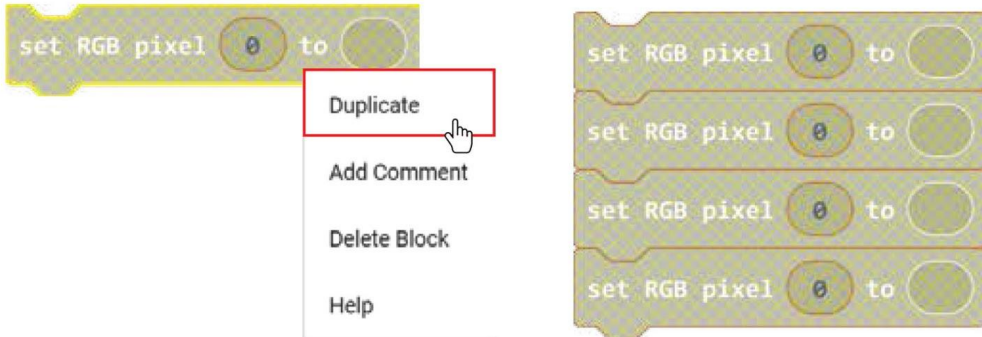
Gyerünk kódolni!

1.lépés A MakeCode Editor-ban, hozzon létre egy új projektet az EDU:BIT kiterjesztésekkel.

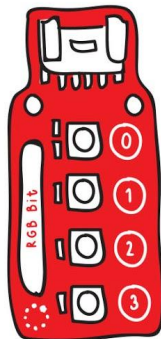
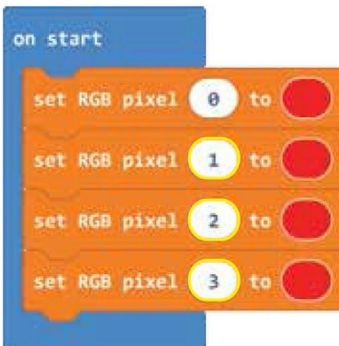
Kattintson [RGB Bit] kategóriára és válassza a [set RGB pixel_to_] blokkot.



2.lépés A munkaterületen, kattintson a jobb gombbal a [set RGB pixel_to_] blokkra és kattintson a 'Duplicate' (Másolásra). 4-szer ismételje meg



3.lépés Helyezze a blokkokat az [on start] blokk nyílásába. Módosítsa az RGB pixelszámot 0-ról 1,2-re és 3-ra a második, harmadik és negyedik blokk esetében.



Az RGB Bit-en 4 RGB LED található, és mindegyikhez van hozzárendelve egy azonosító szám (0-3). Ezzel a számmal programozhatja az egyes LED-eket külön-külön.

10. fejezet : RGB Bit

4.lépés Töltsd fel a kódot az EDU:BIT-re.



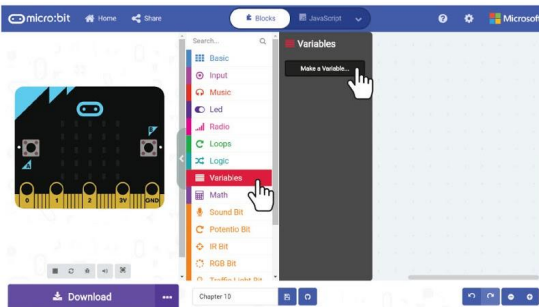
Amikor bekapcsolja a táblát, az RGB Bit négy LED-je kigyullad a beállított színeknek megfelelően.

Könnyedén megváltoztathatja a színt, ha rákattint a blokkra, és kiválaszt egy másik előzetesen színt a palettáról.

Próbáld ki!



5.lépés Adjon két új változót, amiknek a neve- „Right Color and Position” és „Right Color but Wrong Position”.



New variable name:

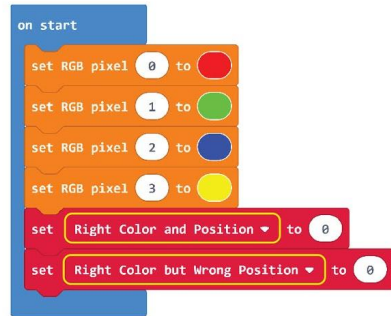
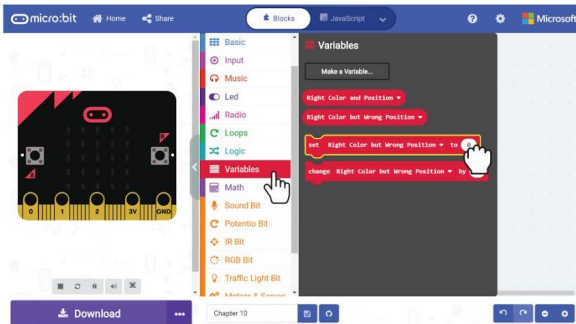
Ok ✓ Cancel ✕

New variable name:

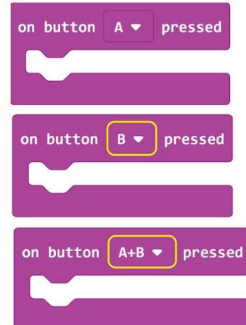
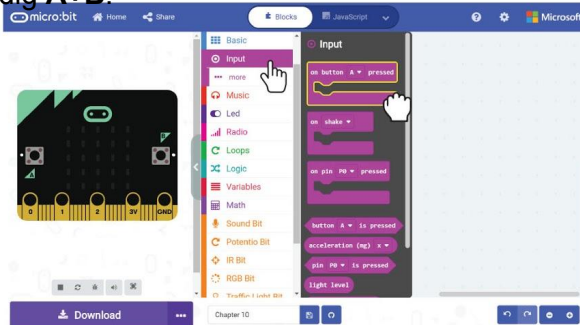
Ok ✓ Cancel ✕



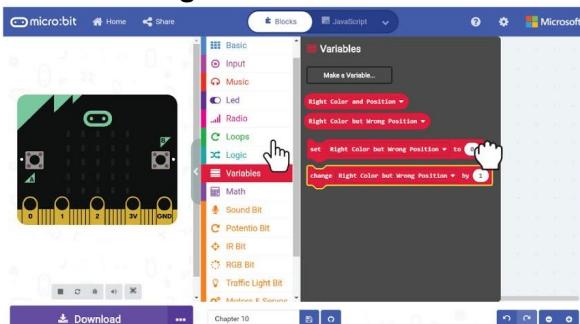
6.lépés Kattintson a **[Variables]** kategóriára és válassza a **[set_to_]** blokkot. Duplázza meg a blokkot, majd helyezze az **[on start]** blokkba. Az első változó legyen a „Right Color and Position”, míg a másik a „Right Color but Wrong Position”.



7.lépés Kattintson az **[Input]** kategóriára és válassza az **[on button_pressed]** blokkot. Készítsen másolatot a blokkról. Változtassa meg a második blokkot **B**-re és a harmadikat pedig **A+B**.

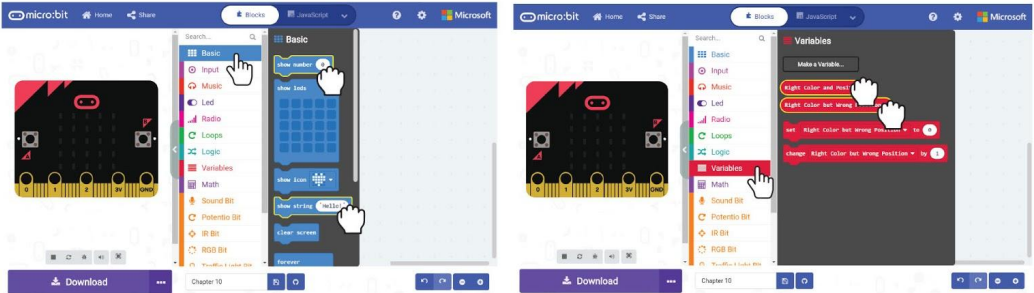


8.lépés Kattintson a **[Variables]** kategóriára újból és válassza a **[change_by_]** blokkot. Másolja és csatolja a blokkokat az **[on button A pressed]** és az **[on button B pressed]** blokkokhoz. Tegye a változókat a „Right Color and Position”, míg a másikat „Right Color but Wrong Position”.



10. fejezet : RGB Bit

9.lépés Adja hozzá a [show number], a [show string] blokkokat a [Basic] kategóriából, valamint a [Right Color and Position] és a [Right Color but Wrong Position] blokkokat a [Variables] kategóriából.



10.lépés Változtassa meg a „Hello” szót a [show string] blokkokban „A=” és „B=” értékekre. Itt a teljes kód.

```
on start
  set RGB pixel 0 to red
  set RGB pixel 1 to green
  set RGB pixel 2 to yellow
  set RGB pixel 3 to blue
  set Right Color and Position to 0
  set Right Color but Wrong Position to 0
```



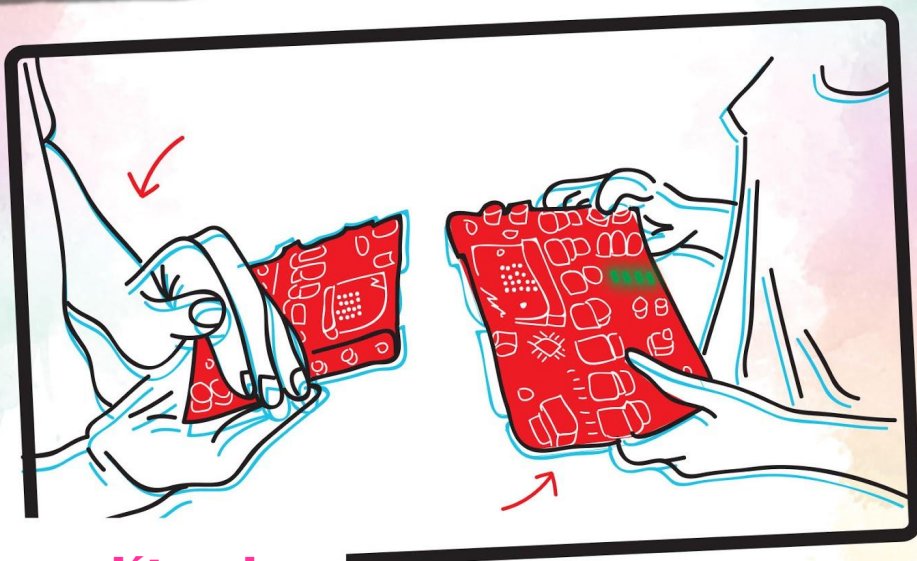
```
on button A pressed
  change Right Color and Position by 1
  show number Right Color and Position

on button B pressed
  change Right Color but Wrong Position by 1
  show number Right Color but Wrong Position

on button A+B pressed
  show string "A ="
  show number Right Color and Position
  show string "B ="
  show number Right Color but Wrong Position
```

11.lépés Töltse fel a kódot az EDU:BIT-re. Most már készen áll a Mastermind Game játékára.

Kezdődjön a játék!



Hogyan játsszuk:

Az 1. játékos, a Kódkészítő, úgy állítja be a titkos kódot, hogy a négy LED-et véletlenszerű sorrendben világítja meg. Csak a kezdethez korlátozza a szín opciót és értékre. Ne felejtse el letakarni a tábláját, hogy a másik játékos ne láthassa a színmintát.

A 2. játékos, a Kódfeltörő, megpróbálja kitalálni a titkos kódot. A 2. játékos felgyújtja az RGB LED-eket az EDU: BIT-en, majd megmutatja a Kódkészítőnek.

A kódkészítő ellenőrzi, majd megnyomja a sárga gombot (A gomb) és / vagy a kék gombot (B gomb) a kódtörő EDU: BIT-en, hogy jelezze, hány LED világít a „right color and position” és a „right color but wrong position”.

Ezután a Kódfeltörő egyszerre megnyomhatja a sárga és a kék gombot, hogy „elolvassa” a visszajelzéseket.

Ismételje meg a 2. és 3. lépést, amíg a Kódfeltörő helyesen kitalálja a színsorozatot (fordulónként legfeljebb 10 kísérlet).

Cseréljen szerepeket, hogy játszhasson egy újabb fordulót.

Hogyan lehet nyerni:

Megnyeri a játékot, ha sikeresen feltörte a kódot (kitalálja a színsorozatot). Ennek elmulasztása esetén a győzelem a Kód Készítőé.



További blokkok

#1 Az RGB Bit LED-jeit úgy állíthatjuk be, hogy a szivárvány színeit megjelenítsék, ha a [set RGB pixel_to_] blokkot [show rainbow on RGB pixels] blokkra cseréljük.

on start

show rainbow on RGB pixels

#2 Futófény-effektust úgy hozhat létre, hogy az [rotate RGB pixels color by_] blokkot egy [forever] blokkba helyezi. Ne felejtse el hozzáadni egy [pause] blokkot a program lelassításához, hogy lássa a hatást. Íme egy példa kód:

on start





set RGB pixel 0 to 
set RGB pixel 1 to 
set RGB pixel 2 to 
set RGB pixel 3 to 

forever

rotate RGB pixels color by 1
pause (ms) 500

#3 A pixeleket egyenként is eltolhatja a [forever] blokkban a [shift RGB pixel color by_] blokk segítségével. Hozzá kell adnia egy [pause] blokkot a program lelassításához, hogy lássa a hatást, amellyel a pixeleket egyenként kikapcsolják. Íme egy példa kód:

on start

set RGB pixel 0 to 
set RGB pixel 1 to 
set RGB pixel 2 to 
set RGB pixel 3 to 

forever

shift RGB pixels color by 1
pause (ms) 500

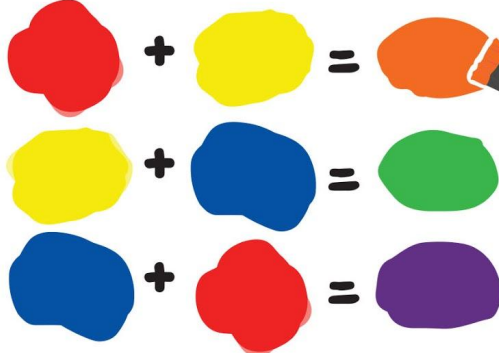


Megváltoztathatja a fenti 2. és 3. effektus irányát, ha a beállítást pozitív értékről negatív értékre változtatja

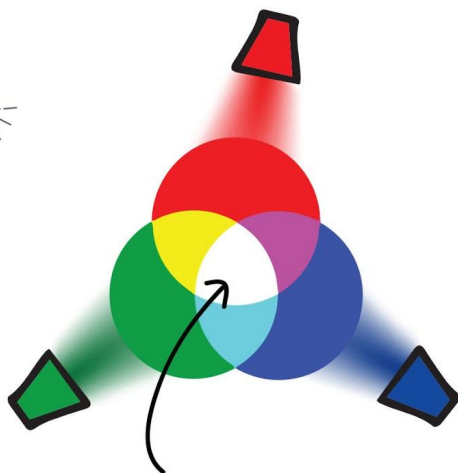
Figyelem!



A művészeti órákon valószínűleg megtudhatja, hogy a 3 elsődleges szín: piros, sárga és kék. És amikor összekevered őket, megkapod ezt az eredményt:



Azonban olyan eszközöknél, amelyek fényeket használnak a színek megjelenítésére, pl. a tv és a számítógép képernyőjén, az RGB színes modellt használják.



Ebben a modellben a három fény a piros, zöld és kék; és kombinálva fehér fényt hoznak létre!

Alkalmazás kihívás

Az EDU:BIT hogyan legyen egy memória játék oktató eszköz.

Hogyan működik?

- Az indításhoz döntse meg az EDU-t: BIT-et balra, hogy az RGB Bit LED-jei véletlenszerű színmintára világítsanak néhány másodpercig.
- A LED-ek kikapcsolása után meg kell figyelnie, majd ki kell mondania a színsorozatot.
- A válasz helyes ellenőrzéséhez nyomja meg a kék gombot (B gomb), hogy az RGB LED-ek ismét ugyanabban a mintában világítsanak.
- Ha helyesen válaszol, nyomja meg a sárga gombot (A gomb) a pontszám frissítéséhez és megjelenítéséhez. Vége a játéknak, ha rosszul válaszolt
- A játék nehézségi szintjét a Potentio Bit gomb forgatásával állíthatja be, hogy növelje / csökkentse a LED-ek világításának időtartamát. A legmagasabb pontszámot elért játékos nyer!
-

Itt van néhány tipp számodra:

Tipp #1 Két változót kell létrehoznia: Score (Pontszám) és Pattern (Minta)

Tipp #2 Az egyes mintákhoz előre be kell állítania a színezési sorrendet (minden RGB LED színe). Használjon több szint vagy előre beállított mintákat, hogy a játék nagyobb kihívást jelenthessen. És fordítva, érdemes korlátozni a fiatalabb játékosok színeit / mintáit.



Bónusz fejezet

Simon beszélő Ledek Radio Communication

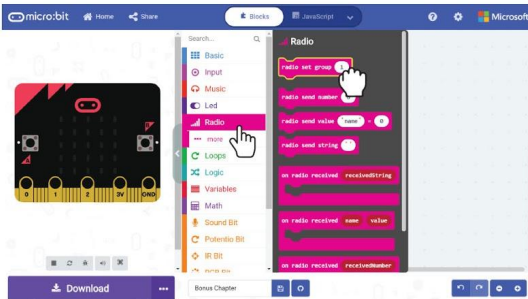




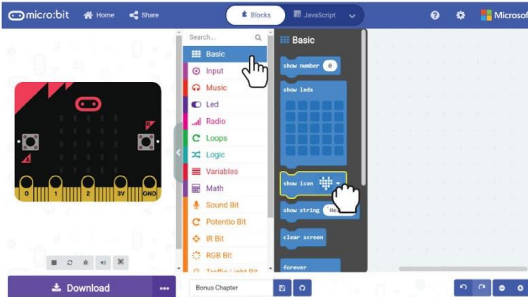
A kommunikáció bármilyen formájához legalább két félre van szükségünk - egy küldőre és egy fogadóra. Ebben a játékban kettő EDU:BIT-re van szükségünk, hogy kommunikálhassunk egymással rádió sugárzási jelek küldésével és fogadásával.

Gyerünk kódolni!

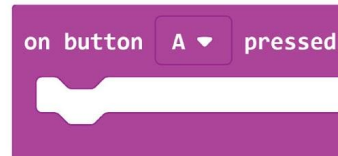
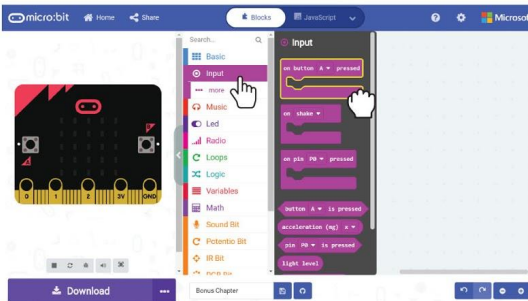
1.lépés A MakeCode Editor-ban, hozzon létre egy új projektet és adja hozzá az EDU:BIT kiterjesztéseit. Kattintson a **[Radio]** kategóriára és válassza a **[radio set group_]** blokkot. Helyezze a blokkot az **[on start]** blokkba.



2.lépés A MakeCode Editor-ban, kattintson a **[Basic]** kategóriára, és adja hozzá a **[show icon]** blokkot a programhoz.



3.lépés Kattintson az **[Input]** kategóriára és válassza az **[on button_pressed]** blokkot.

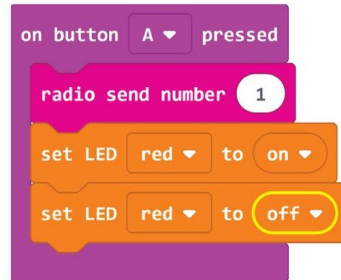
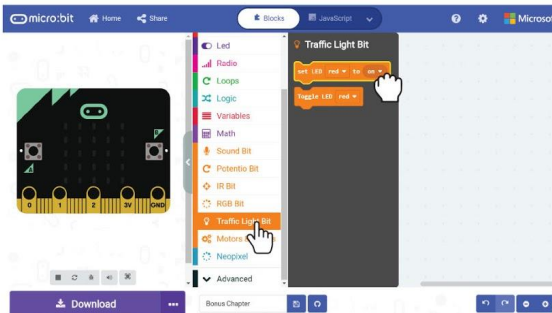




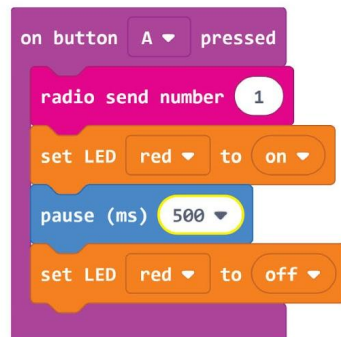
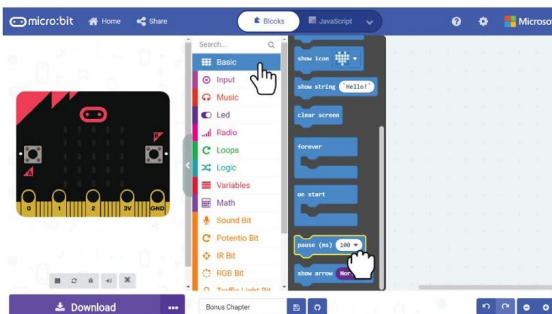
4.lépés Kattintson a [Radio] kategóriára és válassza a [radio send number_] blokkot. Váloztassa az értéket 1-re.



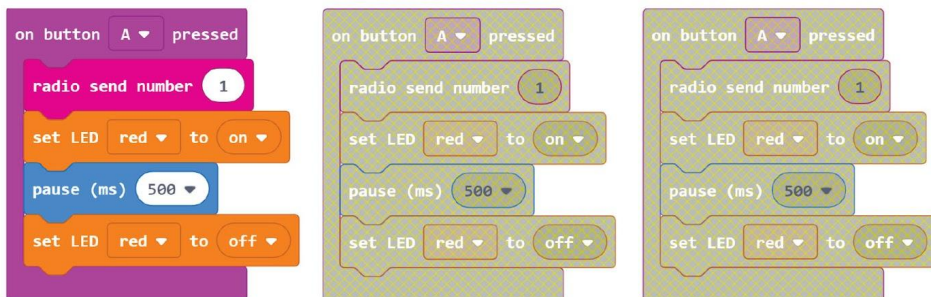
5.lépés Kattintson a [Traffic Light Bit] kategóriára és válassza a [set LED_to_] blokkot. Másolja le, és mindkét blokkot rögzítse az [on button A pressed] blokkba. Módosítsa a második blokkbeállítást 'off'-ra.



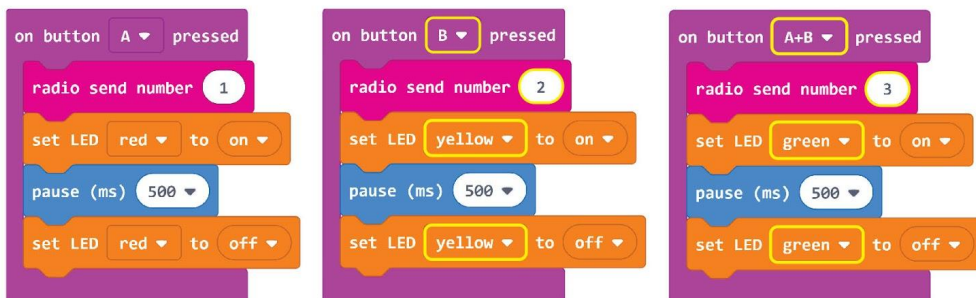
6.lépés Kattintson a [Basic] kategóriára és válassza a [pause(ms)]. Helyezze a blokkot a két [set LED_to_] blokk közé és írja át az értéket 500-ra.



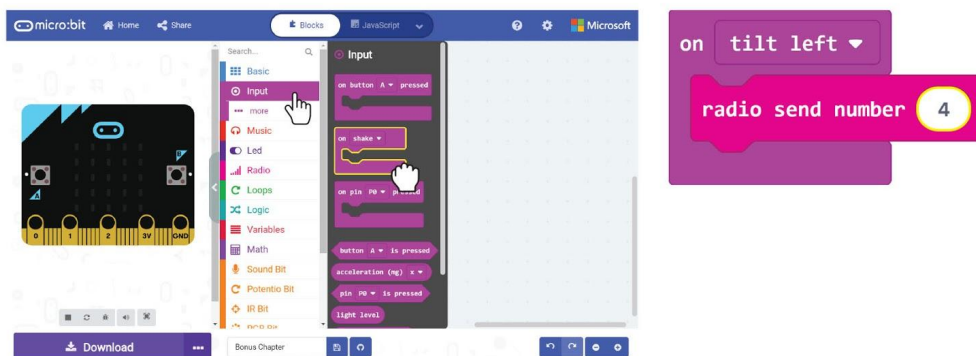
7.lépés Jobb gombbal kattintson az [on button A pressed] blokkra, majd válassza a „Duplicate”(Másolás) lehetőséget. Ismételje meg, hogy 3 blokkot kapjon.



8.lépés Módosítsa a duplikált kódblokkok gombjának, számértékének és LED-színének beállításait a következőkre:

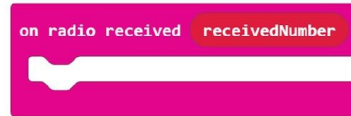


9.lépés Kattintson az [Input] kategóriára és válassza az [on_] blokkot. Módosítsa a beállítást „tilt left”-re. Kattintson a [Radio] kategóriára és válassza a [radio send number_] blokkot. Változtassa meg az étéket 4-re.

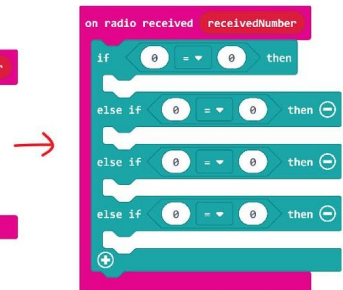
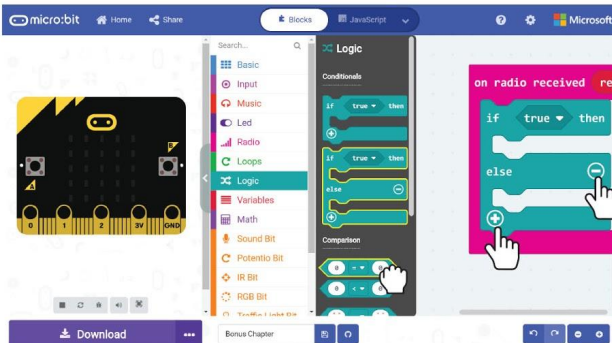




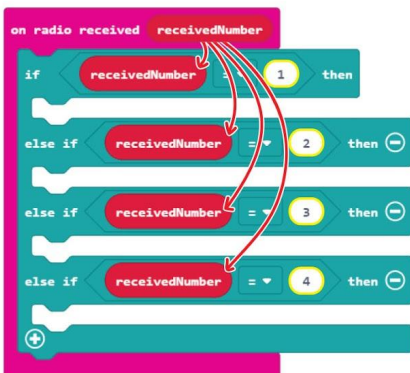
10.lépés Kattintson a [Radio] kategóriára és válasszuk [on radio received receivedNumber] blokkot.



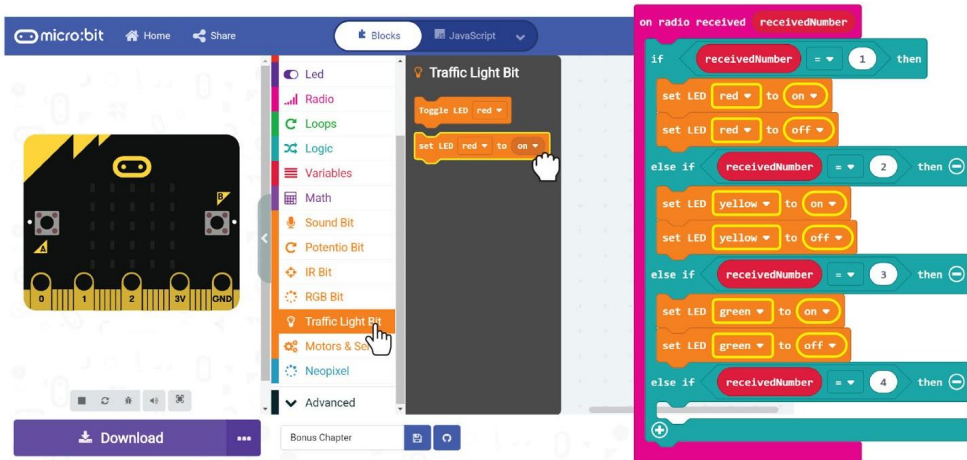
11.lépés Kattintson a [Logic] kategóriára és válassza az [if-then-else] blokkot. Kattintson a + gombra és adjon hozzá három [else if] kiterjesztést és kattintson a – gombra, hogy eltávolítsa az [else] kiterjesztést. Csatolja a [Logic]:[_ = _] összehasonlító blokkokat az „if” és az „if else” mindegyikéhez.



12.lépés Kattintson és húzza a 'receivedNumber' változót a bemutatott összehasonlító blokkokba, és módosítsa az értékeket 1,2,3-ra, illetve 4-re.

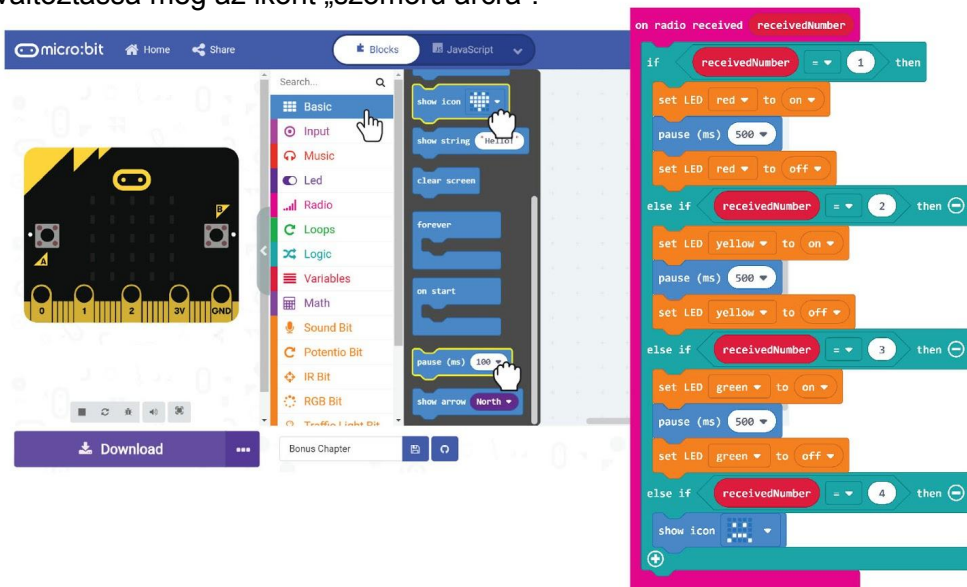


13.lépés Kattintson a [Traffic Light Bit] kategóriára és válassza a [set LED_to_] blokkot. Ismételje meg a blokkot, és rögzítse az első három nyíláshoz. Módosítsa a szint és a be / ki beállítást az ábra szerint.



14.lépés Kattintson a [Basic] kategóriára és válassza [pause (ms)] . Ismételje meg és illessze be a blokkokat a [set LED_to_] blokkok közé, és módosítsa az értéket 500-ra.

Adja hozzá a [Basic]:[show icon] blokkot az utolsó „else if” nyíláshoz és változtassa meg az ikont „szomorú arcra”.





15.lépés Kattintson a [Music] kategóriára és a [start melody_repeating_] gombra a kód kitöltéséhez. Változtassa meg a dallamot „wawawaaa” (vagy tetszőleges dallamot).

Itt van a teljes kód:

Szórás jelek küldéséhez vagy fogadásához az összes "kommunikáló" EDU: BIT-et azonos rádiócsoporthoz kell állítani. Bármely szám lehet 0 és 255 között

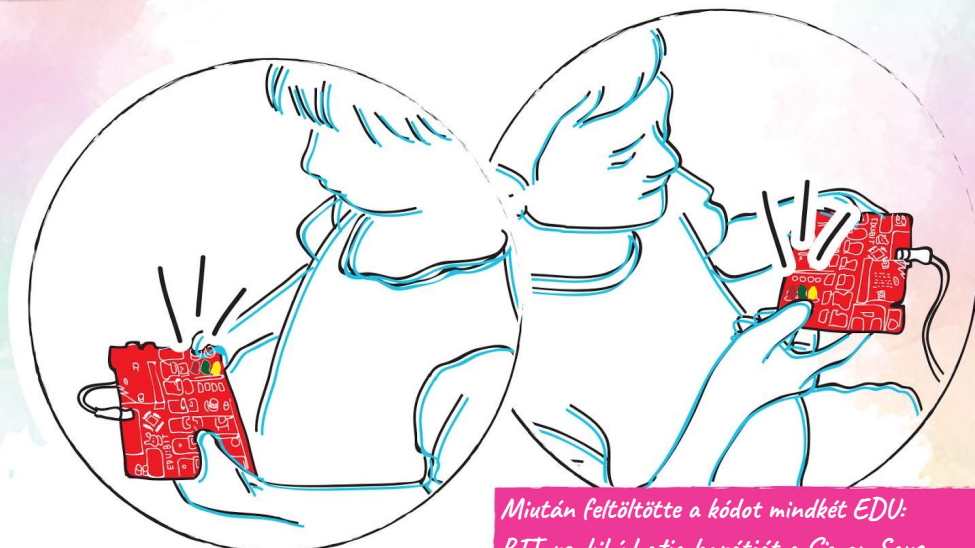
Ezek eseményblokkok. Aktiválásakor az EDU: BIT elküldi a megfelelő parancsot (azaz 1, 2, 3 vagy 4) a közeli EDU: BIT-eknek, amelyek ugyanarra a rádiócsoporthoz vannak beállítva.

16.lépés Töltse fel a kódokat nem csak a saját EDU:BIT-jére, hanem a barátjára is.



Ha mindkét EDU: BIT-be van kapcsolva, rádiójeleket küldhet, hogy felkapcsolja a LED-eket a barátja EDU:BIT-jén; és fordítva, a barátod is megnyomhatja az EDU: BIT gombjait, hogy felkapcsolja a táblán lévő LED-eket.

Kezdődjön a játék!



Hogyan játszunk:

Miután feltöltötte a kódot mindkét EDU: BIT-re, kihívhatja barátját a Simon Says Game interaktív változatára.

Mindkét játékos felváltva "Simon". Nyomja meg a gombokat a piros, sárga és zöld LED-ek kigyulladásához, amikor sorra kerül.

A játék megkezdéséhez az 1. játékos felkapcsolja az egyik LED-et a TrafficLight Bit-en.

A 2. játékos megfigyeli, majd felkapcsolja ugyanazt a LED-et, majd egy másik LED-et.

A játék úgy folytatódik, hogy minden játékos felváltva ismétlje meg a legfrissebb sorrendet, majd felkapcsol egy újabb LED-et, hogy hozzáadja a sorrendhez.

Ha bármelyik játékos rossz LED-et világít (vagy rossz sorrendben), akkor az ellenfél megdönti EDU: BIT-jét a játék befejezéséhez.

A vesztesnek ezt követően vissza kell állítania EDU: BIT-jét egy új játék indításához.

Ez a játék könnyen indul, de minden forduló után egyre hosszabb és összetettebb lesz. A nyereséghez gondosan be kell tartania a sorrendet. Ez egy szórakoztató játék a memória erejének fejlesztésére.





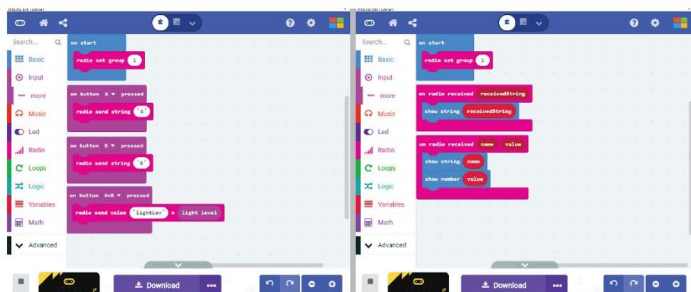
További blokkok

#1 A számok küldése mellett szöveges üzeneteket is küldhet a [radio send string "_"] blokk segítségével. Sztring sugárzási jel fogadásához használnia kell a [on radio received receivedString] blokkot. A karakterlánc maximális hossza 19 karakter.

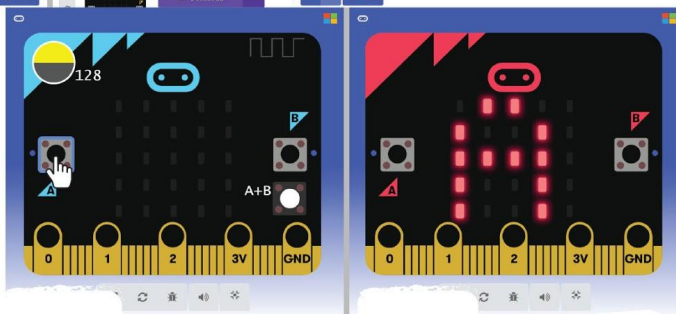
#2 Használja a [radio send value " _ " = _] blokkot és [on radio received name value] blokkot, ha szöveget és számot kell küldenie és/vagy fogadnia. A karakterlánc maximális hossza 8 karakter.



Ha nincs hozzáférése több EDU: BIT-hez, akkor is kipróbálhatja a rádiós kommunikációs funkcióit. Csak keresse fel a makecode.com/multi webhelyet, hogy megírja a "küldő" és a "vevő" kódját. Az eredményt a szimulátor ablakain tekintheti meg.



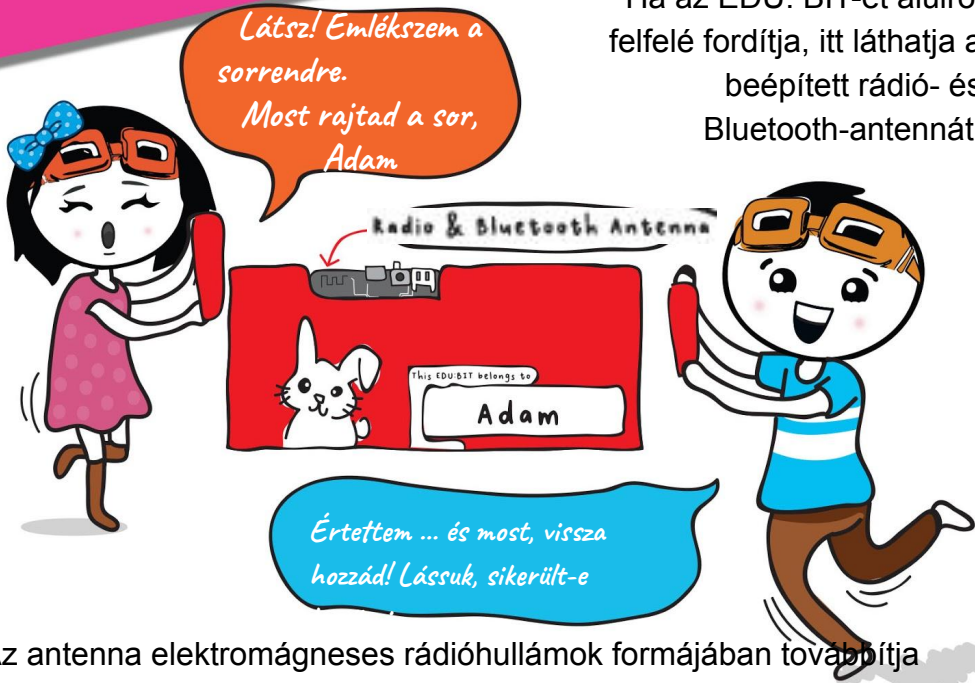
Kattintson a micro: bit táblára a teljes képernyős megjelenítéshez.



Amikor megnyomja az "A" gombot a "küldő" micro: biten, a vevő a micro: biten megkapja a rádiójelet és megjeleníti a fogadott karakterláncot, azaz A. Mi történik, ha megnyomja az A + B gombokat?

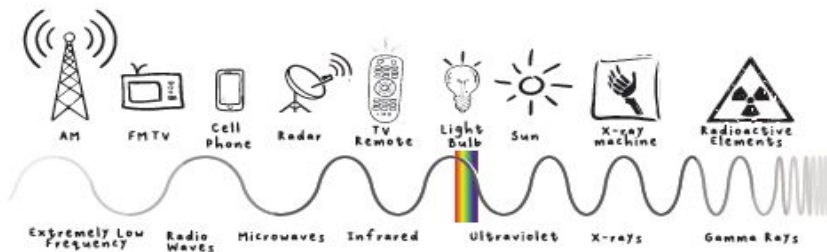


Figyelem!



Az antenna elektromágneses rádióhullámok formájában továbbítja a jeleket, amelyeket televíziós és rádiós műsorszóráshoz, valamint műholdas adásokhoz is gyakran használnak.

Electromagnetic Spectrum



Megjegyzés!




Annak érdekében, hogy az EDU: BIT rádió sugárzási jeleket küldhessen és fogadjon más EDU: BIT-ekből, mindegyiket ugyanahhoz a rádiócsoporthoz kell állítania.

Alkalmazás kihívás

Állítson be visszacsatoló hálózati kommunikációs rendszert az osztályához.

Hogyan működik?

Az osztály minden EDU: BIT-jét állítsa ugyanarra a rádiócsoporra. A tanár EDU:BIT-je be van állítva, hogy gördítsen szöveget, amikor "string" jelet kap, és kigyulladnak a LED-ek a Traffic Light Bit-en, amikor "szám" rádiójelet kap, ezáltal...

Beérkezett szám	Világító Led	Mit jelent?
1		A/Nem/Hamis
2		B/Talán/Nem biztos
3		C/Igen/Igaz

A diákok EDU:BIT-jei úgy vannak beállítva, hogy küldjenek egy karaktersorozatot a tanuló nevével, majd elküldjenek egy számot (1 vagy 2 vagy 3) a tanár EDU: BIT-jén lévő LED-ek felgyulladásához.

Nyomja meg az A gombot az 1. szám elküldéséhez.

Nyomja meg a B gombot a 2. szám elküldéséhez.

A 3. szám elküldéséhez nyomja meg az A + B gombokat.

Itt egy tipp. Rendeljen rövid (csak két vagy három karakteres) beceneveket minden tanulóhoz a szöveggörgetési idő csökkentése érdekében.



Megtanultam...



- az EDU:BIT-en szöveget és animációt megjeleníteni a LED mátrixon.
- letölteni, elmenteni, közzé tenni és szerkeszteni a MakeCode.hex fájlokat



- bemeneti blokkokat használni eseményalapú programozáshoz.
- változókat létrehozni és használni.



- piezzo hangjelzőt használni a Music Bit-en, egyszerű dallamokat lejátszani
- funkciókat létrehozni és használni.
- zenét pontozni



- az EDU:BIT közlekedési lámpa működését
- bővítményeket hozzá adni a MakeCode Editorban.



- objektumokat felismerni az IR Bit segítségével.
- használni a ciklusokat
- tömböket létrehozni és használni.



- az EDU:BIT-en az analóg értékeket leolvasni a Potentio Bit-ről.
- feltérképezni az analóg bemeneteket.
- logikai blokkokat használni a feltételes programozáshoz.



- az EDU:BIT-en keresztül a Sound Bit, azon belül is a hangokat észlelni.
- az eseményindítókkal váltani több mód között.



- az EDU:BIT DC motorját használni- forgási irányt és sebességet szabályozni.
- matematikai blokkok segítségével számtani műveleteket elvégezni.



- az EDU:BIT-en szervomotor -aguláris helyzetének vezérléséhez.



- az RGB ledeket használni az RGB Bit-en, különböző színeket/ mintákat felhasználni.



- rádiójeleket fogadni és küldeni az EDU:BIT-ről.

GRATULÁLUNK!!!

Végig csinálta az összes fejezetet, és megtanult kódolni a MakeCode Editor segítségével. Reméljük, hogy jól érezte magát és kijátszotta a gyermekkori játékokat. És remek, amiért kifejlesztett néhány praktikus alkalmazást az osztályterme számára.

Mostanra már meg kellett értened mindent, mit csinál a micro: bit és az összes extra bit mellett, amelyek az EDU: BIT táblához érkeznek. Pszt ... tudod, hogy letörheted a Biteket?

Menj, és "törd" őket, ha úgy tetszik. Miután letörted a fő táblától, új projekteket készíthet a különféle bitekkel; ezeket a mellékelt plug-and-play kábelekkel kell csatlakoztatnia.

És most itt az ideje, hogy elgondolkodjon és ötleteljen az új játékok és alkalmazások számára. Alig várjuk, hogy lássuk, milyen elképesztő projektekkal fog előállni.

Ne felejtse el megosztani velünk alkotásait. Küldjön e-mailt, vagy hagyjon üzenetet az FB oldalunkon. Szeretnénk hallani felőled.

Egészségedre,
Ádám és Anna



Tájékoztass majd a fejlődésről!



[link.cytron.io/
edubit-resource-hub](https://link.cytron.io/edubit-resource-hub)



