5. alkalom (mozgás, gesztusok)

Tematikai egység	Alkalmazott módszerek, munkaformák	Időtartam
A gyorsulásmérő használata	Frontális tanári magyarázat, közös programírás	15 perc
Gyorsulás és zene	Közös programírás	10 perc
A gesztus fogalma, gesztusok fajtái	Frontális tanári magyarázat	10 perc
A 8-as golyó	Közös programírás	20 perc
A többi gesztus kipróbálása	Egyéni programírás	15 perc
A 8-as golyó program módosítása	Egyéni programírás	10 perc

1. A gyorsulásmérő használata

A micro:biten található egy beépített gyorsulásmérő is, ami három tengely mentén képes mérni az elmozdulást. Ezek a következők:

- X – balra és jobbra döntés
- Y – előre és hátra döntés
- Z – felfelé és lefelé mozgatás

Minden tengelyhez tartozik egy metódus, ami visszaad egy pozitív, vagy egy negatív számot, ami a mért érték, ezred g-ben. Amikor a mért érték 0, az a tengely „egyenesben” van. Példaként a következő program kiírja a micro:bit kijelzőjére, hogy merrefelé dől az eszköz:

```
1. while True:
2.     reading = accelerometer.get_x()
3.     if reading > 20:
4.         display.show("J")
5.     elif reading < -20:
6.         display.show("B")
7.     else:
8.         display.show("-")
```

A második sorban leolvassuk az X tengelyen mért értéket, ahol ezt el is tároljuk egy változóba. (A példához hasonlóan létezik a `get_y()` és a `get_z()` metódus is.) Ezután

ettől az értéktől függően kiírjuk, egy-egy betűvel, hogy éppen merre dől az eszköz. Az elmélet alapján elég lenne azt ellenőriznünk, hogy a mért szám 0-nál kisebb, vagy nagyobb-e, azonban ez túl érzékeny volna, gyakorlatilag lehetetlen lenne középen tartani az eszközt, ezért egy kicsit nagyobb határt hagytunk az egyenesen tartásnak. Ha kipróbáljuk a programot így is látni fogjuk, hogy nem olyan egyszerű a kötőjelet a kijelzőn tartani. Mivel azt szeretnénk, hogy az eszköz folyamatosan mérjen és írja ki az éppen aktuális állapotnak megfelelő jelet, ezért az egészet egy végtelen cikluson belülre írjuk.

Kérdezzük meg a diákoktól, hogy honnan lehet ismerős a fenti program működése! Valószínűleg eszükbe fognak jutni az okostelefonok, amik szintén egy ilyen gyorsulásmérő segítségével állapítják meg, hogy éppen merre dől a telefon, hogyan kell tájolniuk a kijelzőt, vagy éppen a játékokban hogyan akarjuk irányítani az autót, szereplőt stb.

Feladat a diákok számára

Készíts programot, ami kiírja a kijelzőre, hogy az eszközt előre, vagy hátra döntjük éppen! Kísérletezz az értékekkel!

Amennyiben nem boldogulnak a tanulók a feladattal, segítségképpen felidézhetjük a téma elején tárgyalt tengelyeket. Kérjük meg a diákokat, hogy emlékezzenek vissza, hogy melyik tengely mentén lehetett mérni az eszköz előre és hátra döntését! Ezek alapján az előző programot néhány helyen módosítva megkapjuk a feladat jó megoldását.

2. Gyorsulás és zene

Eddig mindig csak egy-egy részét használtuk a micro:bitnek. Próbáljuk meg ezúttal az eszköz különféle funkcióit összekapcsolni! Írjunk egy programot, ami a döntés mértékétől függően játszik le hangokat!

```
import music

while True:
    music.pitch(accelerometer.get_y(), 10)
```

Az Y tengelyről beolvasott értéket adjuk át frekvenciaként a `pitch()` metódusnak, ami ezt a hangot 10 ms-ig játssza le, hogy követni tudja a változást a döntésben. A végtelen ciklus miatt ismételten folyamatosan futó programot kapunk. Teszteljük le, hogy

bizonyos mértékű döntéseknél milyen hangot hallunk (ha hallunk egyáltalán)! Ehhez természetesen össze kell kötnünk az eszközöket egy-egy fülhallgatóval.

3. A gesztus fogalma, gesztusok fajtái

Az előző részben megismert gyorsulásmérő „hozadéka” az, hogy a micro:bit különféle gesztusokat is tud érzékelni. Ez azt jelenti, hogy ha az eszközt egy bizonyos módon mozgatjuk (ez a gesztus), képes érzékelni azt, és a hozzá tartozó utasításokat végrehajtani. A Pythonban a következő gesztusok használatára van lehetőségünk: `up`, `down`, `left`, `right`, `face up`, `face down`, `freefall`, `3g`, `6g`, `8g`, `shake`. A gesztusok neveit ugyanúgy használhatjuk, mint a stringeket. A legtöbb gesztus neve magától értetődő, a `3g`, `6g`, és `8g` gesztusok a névnek megfelelő g erőhatás elérésekor aktiválódnak.

A jelenleg érvényben lévő gesztust az `accelerometer.current_gesture()` metódussal tudjuk lekérdezni. Az eredménye valamelyik fenti gesztus lesz. Nézzük meg egy egyszerű példaprogramon keresztül ennek használatát! Írjunk a fentiek alapján közösen egy programot, ami egy boldog arcot rajzol ki a kijelzőre, hogyha az eszköz felfelé néz, egyébként pedig egy ideges arcot:

```
1. while True:
2.     gesture = accelerometer.current_gesture()
3.     if gesture == "face up":
4.         display.show(Image.HAPPY)
5.     else:
6.         display.show(Image.ANGRY)
```

Végtelen ciklusba csomagoljuk a programot, hiszen folyamatos végrehajtásra van szükségünk. A második sorban lekérdezzük a jelenlegi gesztust a korábban már ismertetett metódussal, majd egy elágazás segítségével eldöntjük, hogy az eszköz felfelé néz-e. Ha igen, a beépített boldog arcot használjuk, ha nem, az ideges arcot.

4. A 8-as golyó

Különböző amerikai filmekben, rajzfilmekben találkozhatunk a varázserejű fekete 8-as biliárdgolyóval, ami minden eldöntendő kérdésünkre megadja a választ. Alakítsuk most a micro:bitet varázserejű 8-assá! Tegyünk fel a micro:bitnek egy kérdést, rázzuk meg, majd olvassuk el a kijelzőn, hogy mi a válasz rá! Írjunk egy programot, ami megvalósítja ezt a működést:

```
1.  import random
2.
3.  answers = [
4.      "Biztosan igy lesz",
5.      "Ketsegkivul",
6.      "Igen, mindenkeppen",
7.      "Szamithatsz ra, hogy igy lesz",
8.      "Valoszinuleg",
9.      "Igen",
10.     "A csillagok szerint igen",
11.     "Kesobb kerdezd meg ujra",
12.     "Most nem mondhatom el",
13.     "Jelenleg megjosolhatatlan",
14.     "Koncentralj, es kerdezd meg ujra",
15.     "Ne szamits ra"
16.     "A valaszom nem",
17.     "A forrasaim alapjan nem",
18.     "Nem tunik valoszinunek",
19.     "Nagyon ketseges",
20. ]
21.
22. while True:
23.     display.show("8")
24.     if accelerometer.was_gesture("shake"):
25.         display.clear()
26.         sleep(1000)
27.         display.scroll(random.choice(answers))
```

A program nagyrészét az előre megadott válaszok listája teszi ki. Természetesen ezeket megalkothatjuk a gyerekekkel közösen is, sőt egymás programjába is beírhatnak általuk kitalált válaszokat, akár úgy, hogy körbe mennek a gépeken, és mindenki ír pár ötletet a másik programjába. Példaként áll itt néhány, idő hiányában ezeket is nyugodtan használhatjuk. A működés a 22. sorban kezdődik. A végtelen cikluson belül, amennyiben megráztuk az eszközt letöröljük a kijelzőt, majd kis „gondolkodási idő” után kiírunk egy véletlenszerűen kiválasztott választ a listából. Ez a 8-as golyó varázsereje! A diákokkal közösen döntsük el az élet nagy kérdéseit, immáron a micro:bitek segítségével!

Feladat a diákok számára

Készíts programot, ami a különböző gesztusokra mind-mind máshogy reagál! Próbáld ki, hogy melyik gesztus mikor lép életbe!

Módosítsd a 8-as golyó programot! Hogyan lehetne „csalni” a golyó varázserejével, hogy kedvünk szerint pozitív, vagy negatív választ adjon? (Tipp: használd a gombokat!)