

LEGO robotok

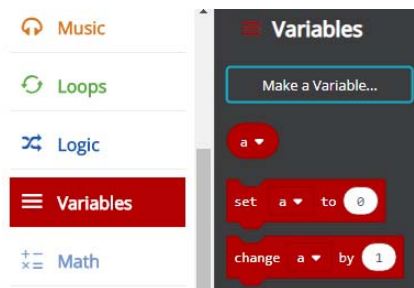
XXIV. rész

Gyakorlatilag ezekből a típusú blokkokból (lásd a XXIII. részben) épül fel a teljes program, az eszköztáron ezen kívül létezik saját használatra létrehozandó blokkoknak is fentartott hely, ezek a Változók és a Függvények.

Ha egy saját változót akarunk létrehozni, akkor a Változók (Variables) lehetőséget kell választanunk, majd innen a Változó létrehozása...

(Make a Variable...) gombot. Ekkor a megjelenő párbeszédablakban meg kell adni a változó nevét. Sem típust, sem semmi más tulajdonságot nem kell megadni, csak egy nevet.

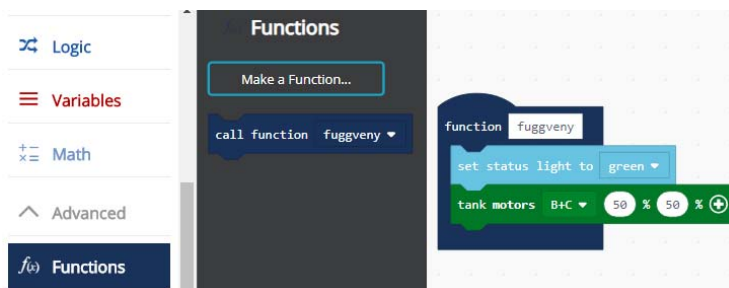
Ha létrehoztunk egy változót (175. ábra), akkor az Eszköztár Változók fülében automatikusan létrejön egy numerikus blokk a változó nevével, valamint két utasításblokk, az egyik a változó értékének beállítására (set), a másik a változó értékének módosítására (change).



175. ábra
Változók

Ha egy új függvényt szeretnénk létrehozni, akkor az Eszköztár Függvények fülében lévő Függvény létrehozása... (Make a Function...)

gombot kell megnyomni. A párbeszédablakban meg kell adni egy nevet, majd a munkaterületen megjelenik egy új lila függvényblokk. Ide helyezhetjük el a függvény utasításait. Paraméterek, visszatérési érték megadására nincs lehetőség, ezeket változók segítségével állíthatjuk be (például a micro:bit programozásánál erre van lehetőség, az EV3-nál viszont nincs). Rekurzív függvények írására sincs lehetőség. A függvény csupán az utasításokat zárja egybe. A Függvények fülben automatikusan létrejön egy függvényhívási utasítás is (176. ábra).



176. ábra
Függvények

A Beállítások gomb segítségével testreszabhatjuk a munkafelületünket. Nevet adhatunk a projektnek (Project Settings), kezelhetjük a kiterjesztéseket (Extensions), kinyomtathatjuk a projektet (Print...), lementhetjük a munkánkat (Save Project), kitörölhetjük azt (Delete

Project), valamilyen visszaélést, hibát jelenthetünk be (Report Abuse), beállíthatjuk a felület nyelvét (Language). Itt jegyezzük meg, hogy míg például a micro:bit esetén 27 nyelv közül választhatunk, köztük a magyar nyelvet is, addig a LEGO EV3 esetén egyelőre csak 5 nyelv érhető el (magyar nincs), a honlap kezelői fel is szólítanak, hogy segítsünk a fordításban.

A fenti beállítások mellett lehetőség van magasabb kontraszt beállítására (High Contrast – On, Off), ekkor a képernyő háttere feketére vált, valamint zöld háttér beállítására (Green Screen – On, Off) is. Ez igen hasznos, ha videót szeretnénk készíteni vagy képeket szeretnénk lementeni.

A Visszaállítás (Reset) menüpont törli az összes projektünket (ennek a törlésnek nincs visszavonás művelete), a Névjegy (About...) menüpont segítségével pedig a LEGO és a MakeCode verziószámát, felhasználási feltételeit tekinthetjük meg.

Ha a Blokk munkaterületen a jobb egérgombbal kattintunk, az előjövő környezetérzékeny menü a következő lehetőségeket kínálja fel:

- Megjegyzés hozzáadása (Add Comment) – segítségével megjegyzés-blokkokat helyezhetünk el a Munkaterületen;
- Az összes blokk törlése (Delete All Blocks) – töröljük a Munkaterületre helyezett összes blokkot;
- A kód formázása (Format Code) – szépen, automatikusan rendezi, elhelyezi a blokkokat;
- Képernyőfotó mentése (Download Screenshot) – képként elmenti a Munkaterületet.

Ha a JavaScript munkaterületen a jobb egérgombbal kattintunk, az előjövő környezetérzékeny menü a következő lehetőségeket kínálja fel:

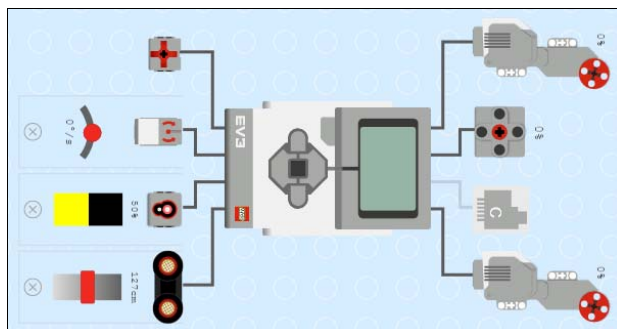
- Mentés (Save) – lementi a dokumentumot (programot);
- Szimulátor futtatása (Run Simulator) – futtatja az EV3 szimulátort;
- Letöltés (Download) – lementi, letölti a programot bináris, futtatható változatban (*.UF2);
- Dokumentum formázása (Format Document) – szépen, automatikusan elrendezi a kódot;
- Kivágás (Cut) – vágólapra helyezi és kitörli a kijelölt szöveget;
- Másolás (Copy) – vágólapra helyezi a kijelölt szöveget;
- Parancs paletta (Command Palette) – könnyen kereshető változatban, egy listában megjeleníti a parancsokat.

Érdekes a szimulátor működése is (177. ábra).

A LEGO MINDSTORMS Education EV3 szimulátor azonnali visszajelzést ad a programozó számára, hogy milyen érzékelők és motorok vannak csatlakoztatva a tégla melyik portjához. Ez egy nagyszerű módszer a programok tesztelésére és a hibakeresésére is.

Megváltoztathatjuk az érzékelők bemeneti értékeit, nyomogathatjuk a gombokat, megnézhetjük a motorok viselkedését, és láthatjuk a kijelzőn megjelenő információkat.

Választhatjuk a teljes képernyős, vagy a nem teljes képernyős működését, a tégla kijelzőjét fel is nagyíthatjuk működés közben.



177. ábra: A szimulátor

III.3.5.3. Programozás MakeCode segítségével

A MakeCode felület a LEGO téglá programozását eseményorientáltan, eseményvezérléssel oldja meg.

Az eseményvezérelt programozás lényege, hogy a teljes program vagy ennek részei, ágai nem szekvenciálisan, előre meghatározott sorrendben futnak le, hanem a vezérlést bizonyos külső vagy belső események határozzák meg, indítják el.

A program így nem más, mint az események bekövetkezésére válaszul végrehajtott eljárások (úgynevezett *eseménykezelők*) halmaza, amelyek nagyrészt egymástól függetlenül dolgoznak.

Az eseményvezérelt architektúrák ez utóbbi okból kifolyólag rendkívül robusztusak és könnyen átláthatók.

A *külső események* a felhasználói bevitel (gomb, billentyű lenyomása, egérművelet, menü kiválasztása stb.), valamint a hardvertől származó események (például az időzítőáramkor, vagy periféria által kiváltott megszakítás, visszahívás).

A *belső események* a program más részeitől vagy más programoktól kapott üzenetek.


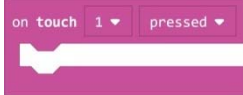
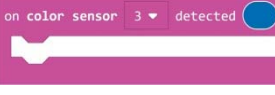
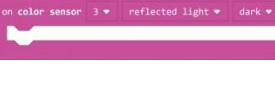
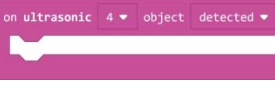
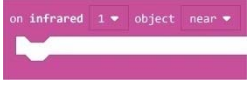
Az események mindig egy komponenshez kapcsolódnak, ahhoz, amelyen ezek bekövetkeznek. Például, ha az egérrel egy gombra kattintunk, akkor az esemény a gomb komponensen keletkezik. Ez egy objektum, és ezt nevezzük az *esemény forrásobjektumának*.

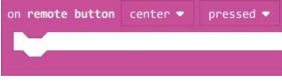



Programozás során természetesen nem minden eseményre kell reagálnunk, csak azokra, amelyek a program szempontjából elengedhetetlenek. Így a programozó mondja meg, hogy melyek azok az események, amelyek bekövetkezésére egy forrásobjektumon a programnak reagálni kell, vagyis a programozó készíti el az egyes eseményekhez az eseménykezelőket.

Az eseményvezérelt programok esetén a főprogram egy ciklusban figyeli a bekövetkezett eseményeket (szekvenciálisan), begyűjti, majd szétosztja (dispatch) ezeket. Így hívódnak meg az eseménykezelők.

A működés során maguk az eseménykezelők is válhatnak további események kiváltójává és forrásává.

A MakeCode eseménykezelőit a 28. táblázat foglalja össze. A portokat itt is ugyanúgy be kell állítanunk, mint LEGO MINDSTORMS EV3 Home Edition esetén. Érdekesség, hogy a portok JavaScript-ben nem paraméterként jelennek meg, hanem a függvények nevében (például `touch1`, `color3` stb.).

Eszköz	Blokk	JavaScript	Jelentés
Tégla		<code>brick.buttonEnter. onEvent(ButtonEvent.Pressed, function () { })</code>	Ha lenyomjuk, felengedjük a tégla valamelyik gombját.
Érzékelők		<code>sensors.touch1. onEvent(ButtonEvent.Pressed, function () { })</code>	Ha az érintés-érzékelőt lenyomjuk, felengedjük.
		<code>sensors.color3. onColorDetected(ColorSensorColor. Blue, function () { })</code>	Ha a színérzékelő valamilyen színt érzékel.
		<code>sensors.color3. onLightDetected(LightIntensity- Mode.Reflected, Light.Dark, function () { })</code>	Ha a színérzékelő valamilyen fényerősséget érzékel.
		<code>sensors.ultrasonic4. onEvent(UltrasonicSensorEvent. ObjectDetected, function () { })</code>	Ha az ultrahang érzékelő valamilyen tárgyat érzékel.
		<code>sensors.infrared1. onEvent(InfraredSensorEvent. ObjectNear, function () { })</code>	Ha az infravörös érzékelő valamilyen tárgyat érzékel.

Eszköz	Blokk	JavaScript	Jelentés
		<code>sensors.remote-ButtonCenter.onEvent(ButtonEvent.Pressed, function () {</code>	Ha lenyomtuk, felengedjük a távirányító valamelyik gombját.
Ciklusok		<code>forever(function () {</code>	Mindig végrehajtódik, folyamatosan fut.
		Nincs külön függvénye, maga a program.	A program kezdetekor hajtódik végre. Inicializáló rész.
Kontroll		<code>control.onEvent(0, 0, function () {</code>	Amikor egy regisztrált esemény történik.

28. táblázat: A MakeCode eseménykezelői

Először tehát kiválasztjuk a programunkhoz szükséges esemény-kezelőket, ezeket felhelyezzük a Munkaterületre, majd beletesszük a szükséges utasításokat. Ha Blokk nézetben vagyunk, akkor egyszerűen az egérrel behúzzuk, ha JavaScript nézetben vagyunk, vagy beírjuk, vagy behúzzuk az egérrel. Ekkor vigyázunk a zárójelekre! Ha egy blokkot a Munkaterületre húzunk, és lenyomjuk rajta a jobb egérgombot, a megjelenő menüből lehetőségünk nyílik másolatot készíteni a blokkról (Duplicate), megjegyzéssel ellátni a blokkot (Add Comment), törölni a blokkot (Delete Block), valamint segítséget kérni a blokkról (Help).



178. ábra

Blokkok megjegyzéssel való ellátása

JavaScriptben a következőképpen láthatjuk el megjegyzésekkel a forráskódot:

```
// Megjegyzés

vagy:

/**
 * Több soros
 * megjegyzés
 */
```

Kovács Lehel István