

Modul 1

**ROBOTIKA PROSTREDNÍCTVOM POSUNKOVÉHO JAZYKA:
ZABEZPEČENIE PRÍSTUPU A ZAPOJENIA ŽIAKOV
SO ZDRAVOTNÝM POSTIHNUTÍM (NEPOČUJÚCICH ALEBO
NEDOSLÝCHAVÝCH) DO DIGITÁLNEHO SVETA KÓDOVANIA
A ROBOTIKY – ROBOTICS4DEAF**

Projekt č.: 2019-1-PL01-KA201-065123



Erasmus+

Tento projekt bol financovaný s podporou Európskej komisie.
Toto oznámenie vyjadruje len názory autora a Komisia
nenesie zodpovednosť za akékoľvek použitie
informácií v ňom obsiahnutých.



Ako čítať tento dokument

Náš vzdelávací materiál sa skladá z dvoch modulov, toto je konceptová verzia modulu 1. Pre lepšiu predstavu o štruktúre vzdelávacieho materiálu si pozrite kompetenčný rámec: (https://drive.google.com/file/d/1858gQJK052zVsk8ivsI5YEEHq2pNkFW_/view?usp=sharing)
-link k sk prekladom kompetenčného rámca

Modul 1 sa skladá z 2 častí. Každá časť sa skladá z niekoľkých kapitol a podkapitol. Ako vidíte v kompetenčnom rámci, máme dve verzie modulu 1, jednu pre každú vekovú skupinu.

Toto je „plná“ verzia určená pre deti od 15 do 21 rokov.

Okrem výroby vzdelávacieho materiálu v písomnej forme sme sa usilovali (pomocou pokynov v príručke) prispôbiť ho cieľovej skupine nášho projektu.

Majte na pamäti, že ide iba o návrhy; nie sme odborníci v tejto oblasti. Podľa nášho názoru je nemožné pripraviť vzdelávací materiál bez zohľadnenia študentov, ktorých oslovujete.

Inštrukcie: (inými slovami, čo znamenajú všetky tie farby?)

Farba **zelená**: táto časť bude preložená do posunkového jazyka.

Color **modrá**: táto časť bude videom.

Žiadna farba: jednoduchý text.

Pre lepšiu predstavu sme pripravili aj beta verziu jednej podkapitoly. Je tiež pripojený a nazýva sa „beta slučky“.

Čakáme na vašu spätnú väzbu. V prípade akýchkoľvek otázok nás neváhajte kontaktovať.

Obsah

1.	SMARC – SMARt Robotic and Coding	3
I.	Stavanie SMARC	3

II.	Naučte sa, ako urobiť SMARC pohybom.	3
A.	Zoskupenie blokov.....	4
B.	Ako urobiť pohyb SMARCu.....	6
C.	Ukážkové programy.....	8
III.	Koncept kódovania slučky	11
2.	Používanie senzorov so Smarcom	14
I.	Senzory	14
A.	<i>Lego Boost senzor farieb a vzdialenosti</i>	14
B.	Detekcia objektov.....	15
C.	Zisťovanie farieb	17
D.	If/Else bloky:	21
II.	Nasledujúce múry so Smarc	22
A.	Nasledujúce múry.....	22
III.	Nasledujúce riadky so Smarcom	28
A.	Chodte po čiare	28
IV.	Detekcia zvuku pomocou protokolu Smarc.....	34
A.	Reakcia na zvuk	34
B.	Bloky zvukového senzora:	34
V.	Navigácia v Smarc pomocou diaľkového ovládača.....	39
A.	Diaľkové ovládanie	39
B.	Bloky diaľkového ovládania:.....	39
VI.	Používanie premenných so Smarcom	42
A.	Matematika a výpočty.....	42
B.	Bloky operátora:	44
C.	Variabilné bloky:.....	44

1. SMARC – SMARt Robotic and Coding

Cieľom tejto časti je oboznámiť študentov so základnými pohybmi robotov. Študenti si zostavia robota SMARC a naučia sa ho programovať tak, aby vykonával základné pohyby. Naučia sa tiež, ako ho pomocou slučiek naprogramovať na opakované pohyby.

I. Stavanie SMARC

Podrobný sprievodca vytvorením protokolu SMARC (Emphasys).



SMARC Building
Instructions.pdf

II. Naučte sa, ako urobiť SMARC pohybom.

Na fotografii nižšie je zobrazené programovacie prostredie aplikácie.

Ukazuje tiež oblasť, kde sa nachádzajú programovacie bloky. Nájdete tam všetko, čo potrebujete na vytvorenie vlastných programov.

Strávte nejaký čas prehliadaním aplikácie.



Foto 1- Bloková paleta

A. Zoskupenie blokov.

V tejto časti sa dozvieme, ako sú bloky zoskupené podľa ich farby. Poznanie všeobecných kategórií uľahčuje hľadanie správneho bloku podľa toho, čo chceme robiť. V nasledujúcich častiach podrobne preskúame každú kategóriu.

- Pomocou **Yellow (žltých)** blokov môžete riadiť priebeh programu

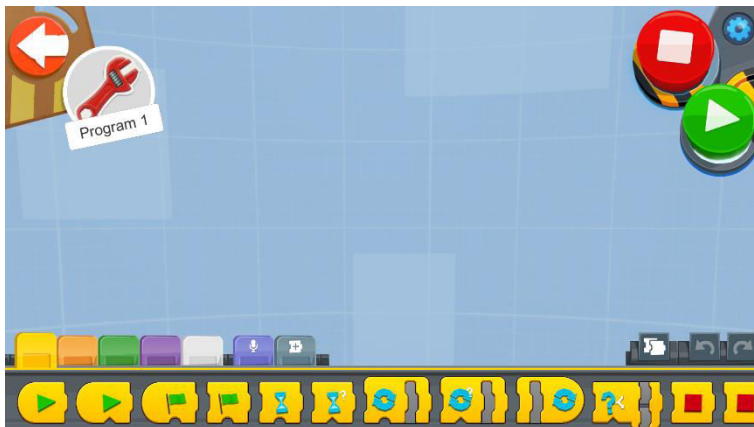


Foto 2-Žlté bloky

Môžete ich použiť na spustenie programu, zastavenie programu, pozastavenie programu alebo dokonca vytvorenie cyklu skupiny príkazov

- **Orange (oranžové)** bloky pracujú so Color and Distance Sensor (snímačom farieb a vzdialenosti), mikrofónom a Move Hub's Tilt Sensor (snímačom náklonu Move Hub)

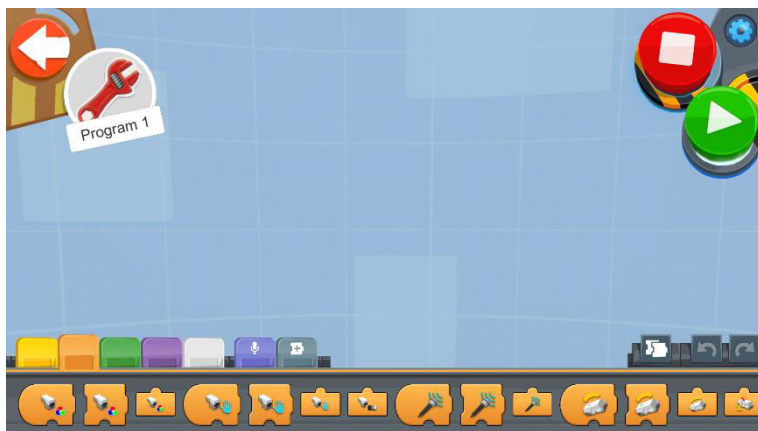


Foto 3-Oranžové bloky

Keď je aktivovaný jeden z vyššie uvedených senzorov, aktivuje sa tiež tieto bloky.

- **Purple (fialové)** bloky sa používajú na komunikáciu robota s vonkajším svetom.



Foto 4-Fialové bloky

Môžu prehrávať zvuky cez reproduktory zariadenia používateľa. Môžu tiež meniť farbu svetiel na zariadeniach Move Hub & Color and Distance Sensor (Move Hub a senzor farieb a vzdialenosti)

- **Biele** bloky vám umožňujú zložité programovanie pomocou “premenných” a “konštánt”.

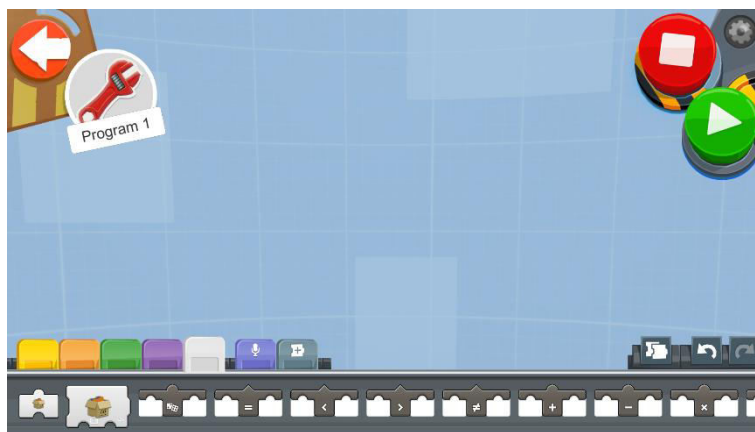


Foto 5-Biele bloky

Môžu byť tiež použité pri vykonávaní rôznych matematických výpočtov, vytváraní logických výrazov a generovaní náhodných čísel

- **Light Purple (svetlofialové)** bloky vám umožňujú zaznamenať zvuk, ktorý má robot opakovať.

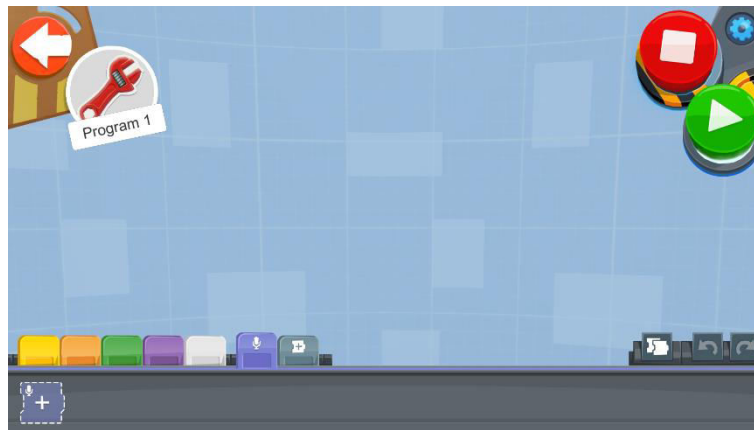


Foto 6-Svetlofialové bloky

- **Group Actions block (blok skupinových akcií)** vám umožňuje zoskupiť ako jedno tlačidlo bloku niekoľko akcií

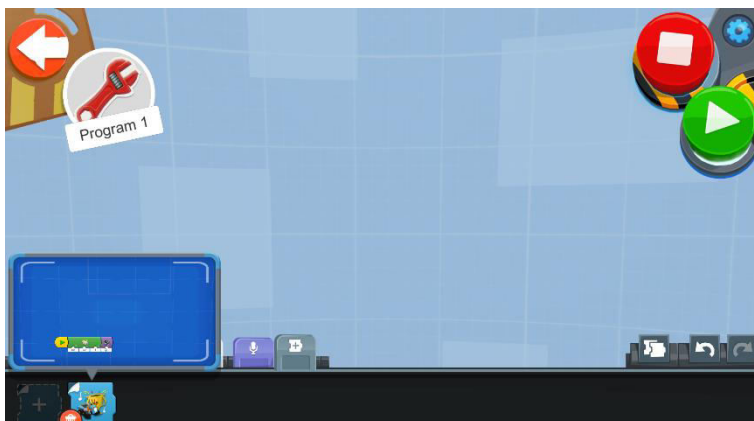


Foto 7- Bloky skupinových akcií

Toto je ekvivalent vytvárania podprogramov

B. Ako urobiť pohyb SMARCu.

Aby sme mohli začať programovať potrebujeme, je StartBlock žltej palety



SMARC, prvá vec, ktorú (štartovací blok) umiestnený v

Existuje niekoľko blokov, ktoré riadia pohyb zariadenia Move Hub. Tieto bloky sa nachádzajú v zelenej sekcii palety. Tu uvádzame tie najbežnejšie.



Move Base for Duration (*presuňte základňu na dobu trvania*) - Motor s jednou rýchlosťou (od -100 do 100) a motor s dvoma rýchlosťami (od -100 do 100) po dobu trvania (v sekundách)



Move Base Tank for Distance (*posuňte základňu nádrže na vzdialenosť*) - motor s jednou rýchlosťou (od -100 do 100) a motor s dvoma rýchlosťami (od -100 do 100) pre vzdialenosť (v stupňoch)



Move Tank and Steer for Duration (*pohybujte nádržou a riadením po dobu trvania*) - otáčky oboch motorov (od -100 do 100) a smer riadenia (od -100 do 100) po dobu trvania (v sekundách)



Move Tank and Steer for Distance (*posuňte nádrž a riadenie na vzdialenosť*) - rýchlosť oboch motorov (od -100 do 100) a smer riadenia (od -100 do 100) na vzdialenosť (v stupňoch)



Move Tank (*pohybujte nádrž*) - motor s jednou rýchlosťou (od -100 do 100) a motor s dvoma rýchlosťami (od -100 do 100)



Move Tank and Steer for Duration (*pohybujte nádržou a riadením po dobu trvania*) - otáčky oboch motorov (od -100 do 100) a smer riadenia (od -100 do 100)

C. Ukážkové programy

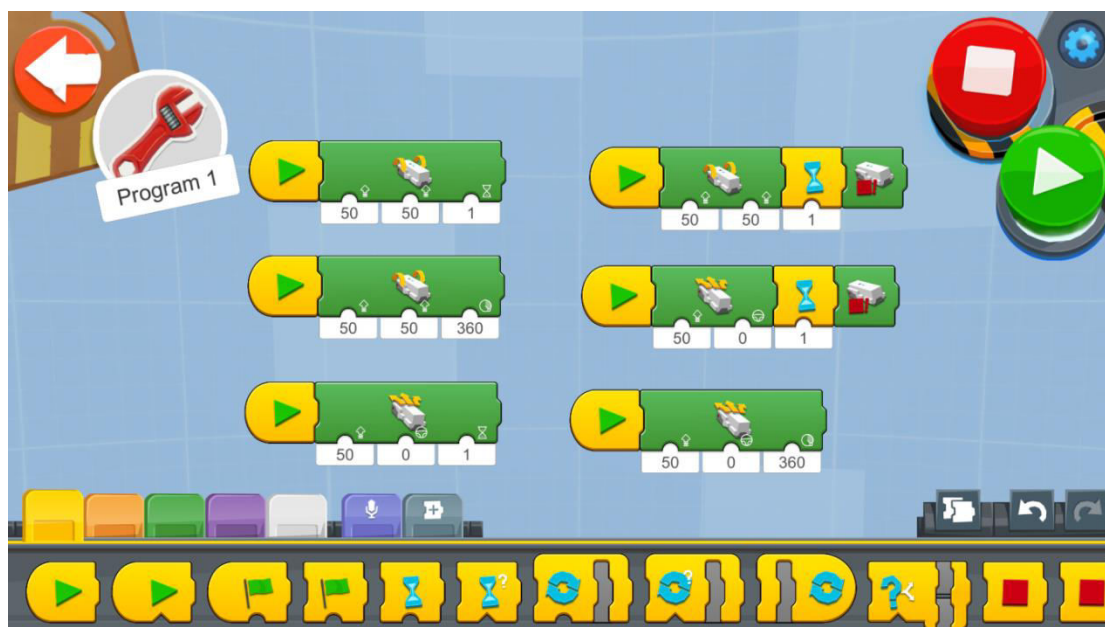


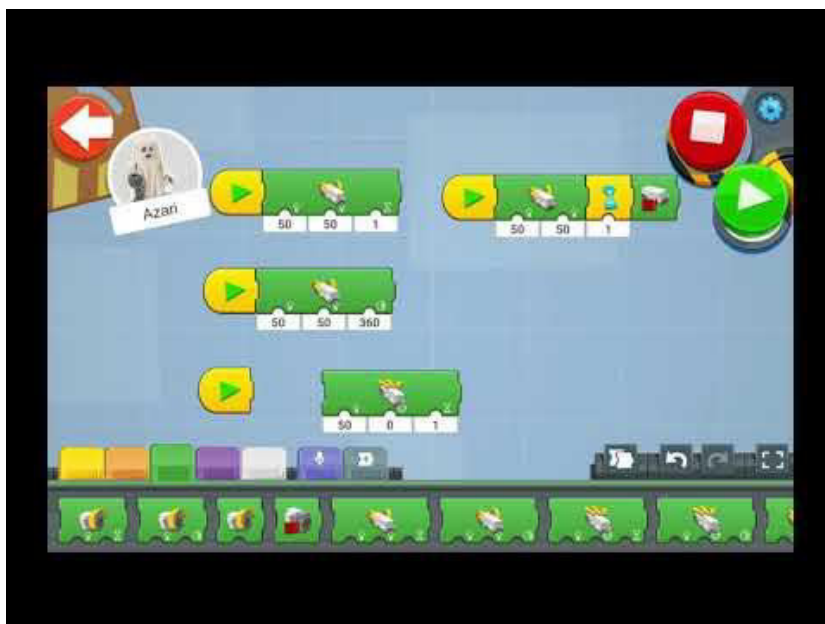
Foto 8 – Tvorba SMARC pohybov.

Skúste použiť rôzne bloky na pohyb ako v príklade vyššie uvedeného programu

Video s nasledujúcim textom a programom Smarc vykonávajúcim programy.

Vytvorte nový projekt na Creative Canva a spustite každý blok kódu a všimnite si rozdiely a podobnosti.

Upozorňujeme, že pohybujúce sa bloky, ktoré neurčujú časový interval alebo počet stupňov (rotácií) pohybu, budú motory otáčať integrovane, pokiaľ nie je pridaný blok čakania, ktorý nastaví časový interval pre prácu motorov, po ktorom nasleduje zastavenie blok motorov.



Naučte sa, ako urobiť Smarc v pohybe. - <https://youtu.be/-Jor9F0y4fA>

Poznámka: ak študent dokončil prvú aktivitu jednoduchým autom alebo prvou zo série Vernies, môže použiť zjednodušeniu verziu pohybových blokov.



Foto 9- Hlavná scéna v hale aplikácie Lego Boost

Krok 1: Prejdite na ikonu Simple Car (*jednoduché auto*) na hlavnej scéne v hale a kliknite na ňu



Foto 10- Aktivita Simple Car (jednoduchého auta) 1

Krok 2: Kliknite na prvú aktivitu

TIP pre učiteľa: Programy môžete odomknúť stlačením aktivity, ktorú chcete odomknúť, na päť sekúnd

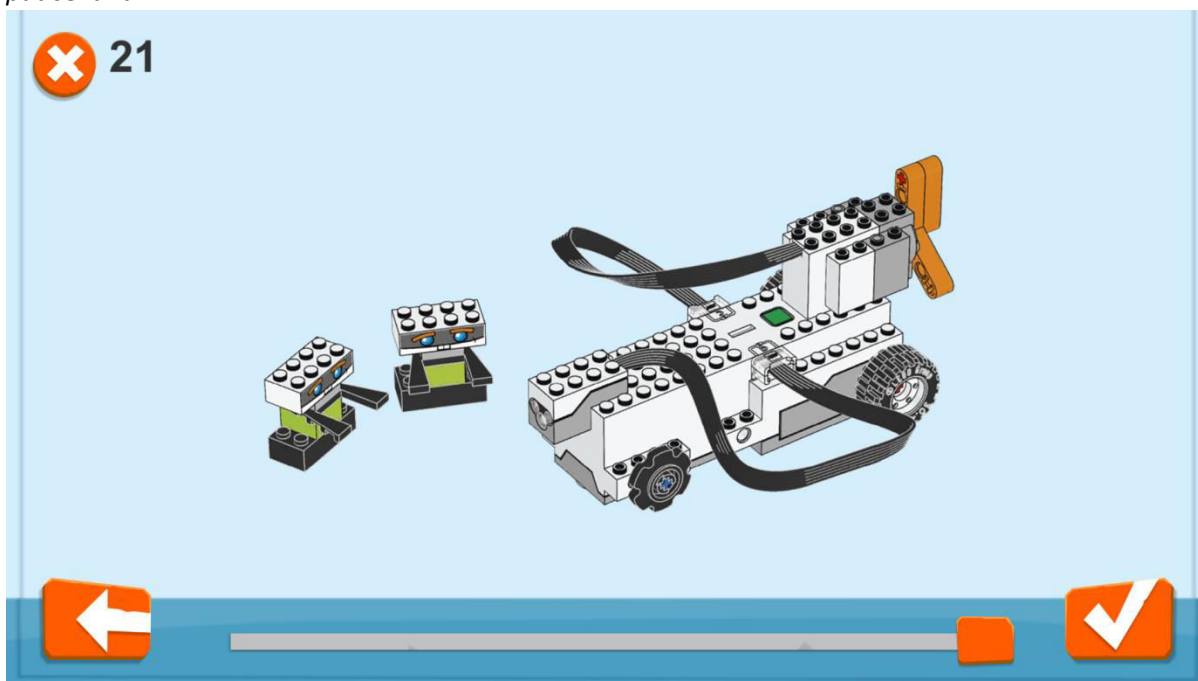


Foto 11- Stavba jednoduchého auta

Krok 3: Ak sa jedná o prvé otvorenie tejto konkrétnej činnosti, ukázal sa podrobný sprievodca pre výrobu automobilu. Vždy sa môžete len posunúť na poslednú stránku a kliknúť na ikonu zaškrtnutia.



Foto 12- Základné bloky pre jednoduché pohyby

Krok 4: Pomocou jednoduchých pohybových blokov pohybnúť Smarcom. Ako ďaleko sa posunie, keď sa použije jeden blok so šípkami? Pohybuje sa očakávaným smerom?

Poznámka pre učiteľa: Hnacie motory pre Smarc sú v prednej časti robota, na vozidle Simple Drive Car (auto s jednoduchým riadením) v zadnej časti robota.

III. Koncept kódovania slučky

Smyčky umožňujú spoločnosti Smarc ľahko opakovať bloky kódu bez toho, aby bolo treba opakovane vkladať bloky kódu. V kódovaní sú rôzne typy slučiek.

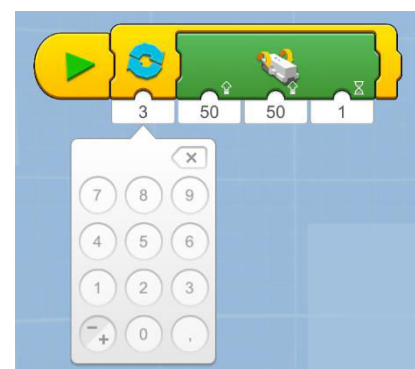
Program „For Loops“ (pre slučky) umožňuje programátorom opakovať kód určitý počet opakovaní.

„While Loops“ (zatiaľ, čo slučky) vyžaduje, aby bola podmienka pravdivá, aby sa mohol opakovať kód.

Program „Forever Loops“ (navždy slučky) opakuje kód navždy a malo by sa mu vyhnúť, ak to nie je potrebné, pretože môže spotrebovať všetky zdroje nášho počítača a vybiť batérie nášho robota.

- **“For Loops” (pre slučky).**

The **Loop For Count** (slučka pre počet): Slučky uzavretých blokov kódu pre konkrétny počet opakovaní, ktoré môže nastaviť užívateľ



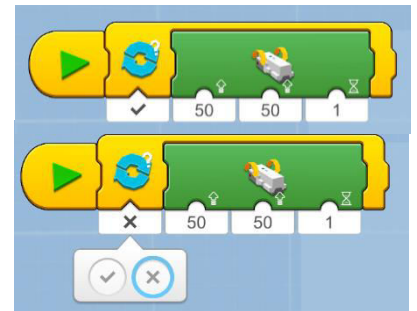
Video

- **“While Loops” (zatiaľ čo slučky...)**

Loop While True (slučka, zatiaľ čo je pravda): Slučky uzavretých blokov kódu, zatiaľ čo podmienka je pravdivá.

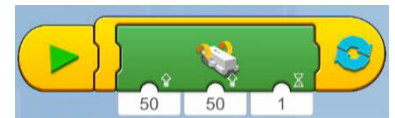
Užívateľ môže nastaviť, či sa slučka vykoná, ak je podmienka pravdivá alebo nepravdivá.

Video



- **“Forever Loops” (navždy slučky)**

Loop Forever (slučka navždy): Slučky priložených blokov kódu



konceptia kódovania slučky - <https://youtu.be/hPyylPF2QZM>

Ukázkové programy blokov slučiek

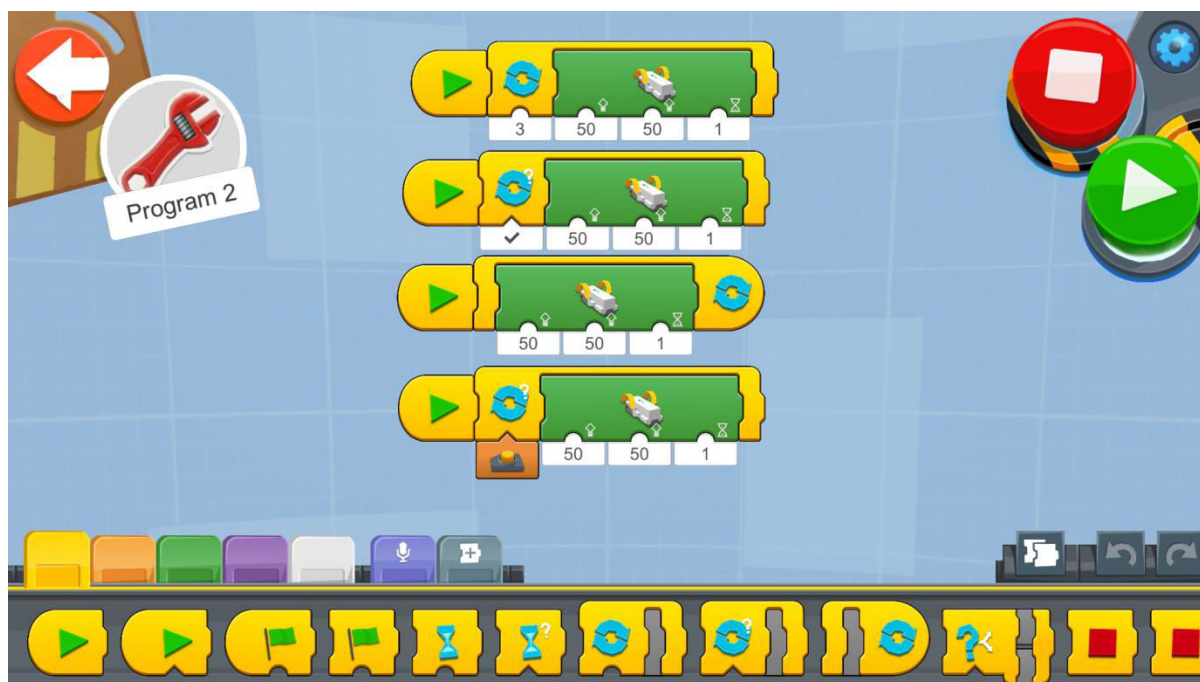


Foto 13-Program – Blok slučiek

Skúste použiť rôzne bloky pre príkazy slučiek, ako je tomu v príklade vyššie.

Vytvorte nový projekt na Creative Canva a spustite každý blok kódu a všimnite si rozdiely a podobnosti.

Všimnite si, že posledný blok kódu nefunguje správne. Vyskúšajte to tak, aby po stlačení tlačidla došlo k aktivácii slučky (podmienka slučky sa stane pravdivou).



slučkové ukážkové programy - <https://youtu.be/PcGk9zD2M5k>

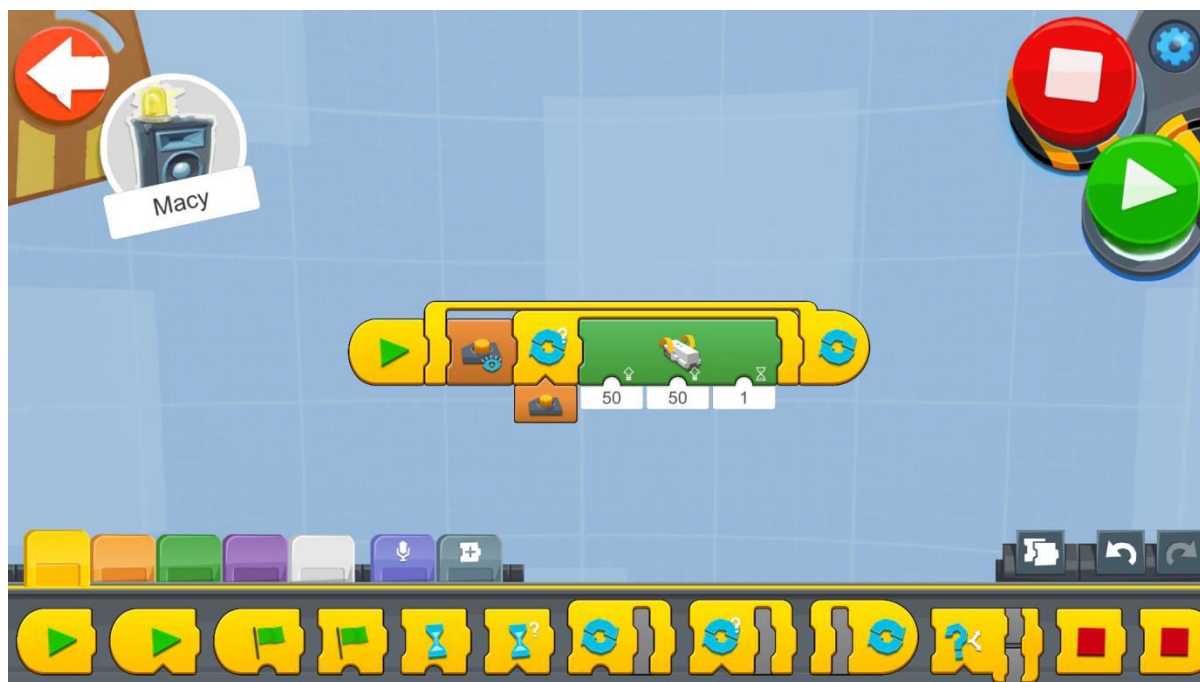


Foto 14-Program – vnorená slučka

Riešením je vložiť ďalšiu slučku Loop Forever (*slučka navždy*) a pridať blok vzhľadu tlačidiel mimo blok Loop While True (*slučka, zatiaľ čo je pravda*). Toto sa nazýva nested loop (*vnorená slučka*) - (slučka vo vnútri slučky).

2. Používanie senzorov so Smarcom

I. Senzory

V tejto časti uvidíme, ako môžeme zabudovať senzor do Smarc a ako môžeme tento senzor využiť na detekciu prekážok, rôznych farieb, intenzity svetla, ako aj vzdialenosti od objektu.

A. *Lego Boost senzor farieb a vzdialenosti*

Senzor je zariadenie, ktoré detekuje udalosti alebo zmeny v okolitom prostredí a odosiela informácie do výpočtového zariadenia, aby ich mohol spracovať a vykonať akcie. V našom každodennom živote existuje veľa príkladov použitia senzorov. Od pohybového senzora, ktorý deteguje pohyb a rozsvieti svetlá alebo otvorí posuvné dvere, až po teplotné senzory, ktoré detegujú teplotu a zapnú ohrievače a podmienky vzduchu. Senzory sa používajú aj vo väčšine typov diaľkovo ovládaných zariadení, napríklad v televízoroch a klimatizáciách.

LEGO BOOST senzor vzdialenosti a farieb, ako je znázornené na obrázku, sa používa na:

- Sensing distance (*snímaná vzdialenosť*) - Ako ďaleko sa nachádza prekážka od snímača.
- Sensing color (*snímajúca farba*) - dokáže zistiť konkrétne farby (Black-čierna, Blue-modrá, Green-zelená, Yellow-žltá, Red-červená, White-biela), ako aj absenciu farby.
- Detecting movement (*detekcia pohybu*) - môže byť použitá ako detektor pohybu
- Detecting light (*detekcia svetla*) - dokáže detekovať úroveň svetla odrazeného od senzora od 0 - najtmavšieho po 10 - najsvetlejšieho.



Senzor farieb a vzdialenosti môže navyše vyžarovať rôzne farby (Red-červená, Green-zelená, Blue-modrá a ich kombinácia troch).

Senzor farieb a vzdialenosti

B. Detekcia objektov

The Detecting Objects Blocks (*blokovanie detekujúcich objektov*)



Trigger on Distance Block (*spúšť na bloku vzdialenosti*) - Spustí sa, keď je vzdialenosť určená snímačom menšia ako vzdialenosť označená hodnotou pod ním. Keď je spustený, vykoná nasledujúcu postupnosť kódu. Hodnoty môžu byť od 0 do 10..



Sensor Distance Reporter (*reportér vzdialenosti snímača*) - Zobrazuje aktuálnu vzdialenosť určenú snímačom v reálnom čase. Musí byť pripojený k spodnej časti ostatných blokov, aby mohol byť zahrnutý do softvéru. Hodnoty môžu byť od 0 do 10.



Wait for Distance (*čakanie na vzdialenosť*) - Čaká na to, aby bola vzdialenosť určená snímačom menšia ako vzdialenosť uvedená hodnotou pod ním. Ak objekt nie je bližšie ako uvedená hodnota, program zostane pozastavený a program začne dodržiavať nasledujúcu postupnosť pokynov, kým nebude splnená podmienka. Hodnoty môžu byť od 0 do 10.

Ukážkové programy zisťovania prekážok

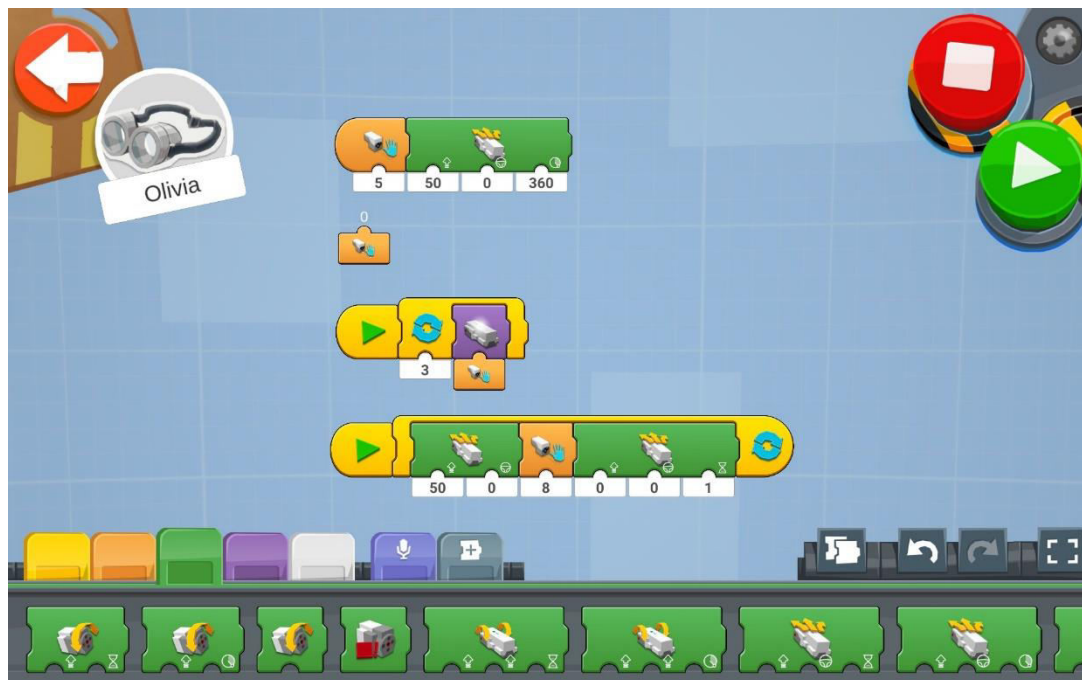


Foto 15 - Zisťovanie prekážok

1. Prvý program sa aktivuje, keď je objekt menej ako vzdialenosť 5 pred Smarcom. Smarc sa potom bude pohybovať o 360 stupňov dopredu rýchlosťou 50.

2. Druhý blok iba indikuje v reálnom čase vzdialenosť od objektu pred snímačom.

3. Tretí program berie vzdialenosť a používa ju ako vstup na výber a zobrazenie farieb na žiarovke movehub. Všimnite si, že keď sa zmení vzdialenosť od objektu, zmení sa aj farba.

Vo štvrtom programe sa Smarc začne pohybovať rýchlosťou 5 dopredu a keď zistí objekt vo vzdialenosti menšej ako 8, na chvíľu sa zastaví, potom sa pokúsi pohnúť znova..



Zisťovanie prekážok - <https://youtu.be/CAXnOmMiUDQ>

C. Zisťovanie farieb

Senzor LEGO BOOST dokáže detekovať šesť farieb (Black-čierna, Blue-modrá, Green-zelená, Yellow-žltá, Red-červená, White-biela a absencia farby - žiadna farba, čo znamená, že senzor nezistí žiadny predmet).

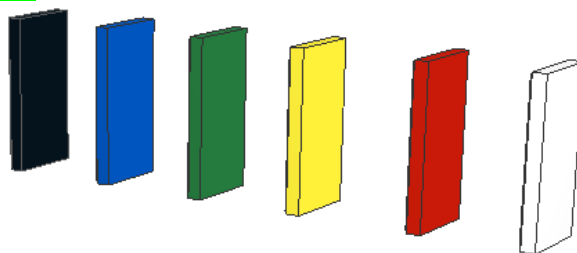


Foto 16- Farby, ktoré senzor vidí: čierna, modrá, zelená, žltá, červená, biela

Poznámky:

Ak sa objekt nenachádza v navrhovanom rozsahu snímača, farba nemusí byť detekovaná správne. Napríklad zelený objekt môže byť detekovaný ako modrý.

Pokiaľ farba objektu nie je jednou zo šiestich detekovateľných farieb, aplikácia ju zobrazí ako najbližšiu možnú farbu. Napríklad oranžová bude prezentovaná ako červená.

The Detecting Colors Blocks (*detekčné bloky farieb*) :



Trigger on Color (*spúšť na farbe*) - Spustí sa, keď sa farba nameraná snímačom rovná farbe znázornenej hodnotou pod ním. Po aktivácii sa vykoná nasledujúca sekvencia kódu. Môže sa použiť sedem hodnôt: žiadna farba, Black-čierna, Blue-modrá, Green-zelená, Yellow-žltá, Red-červená a White-biela.



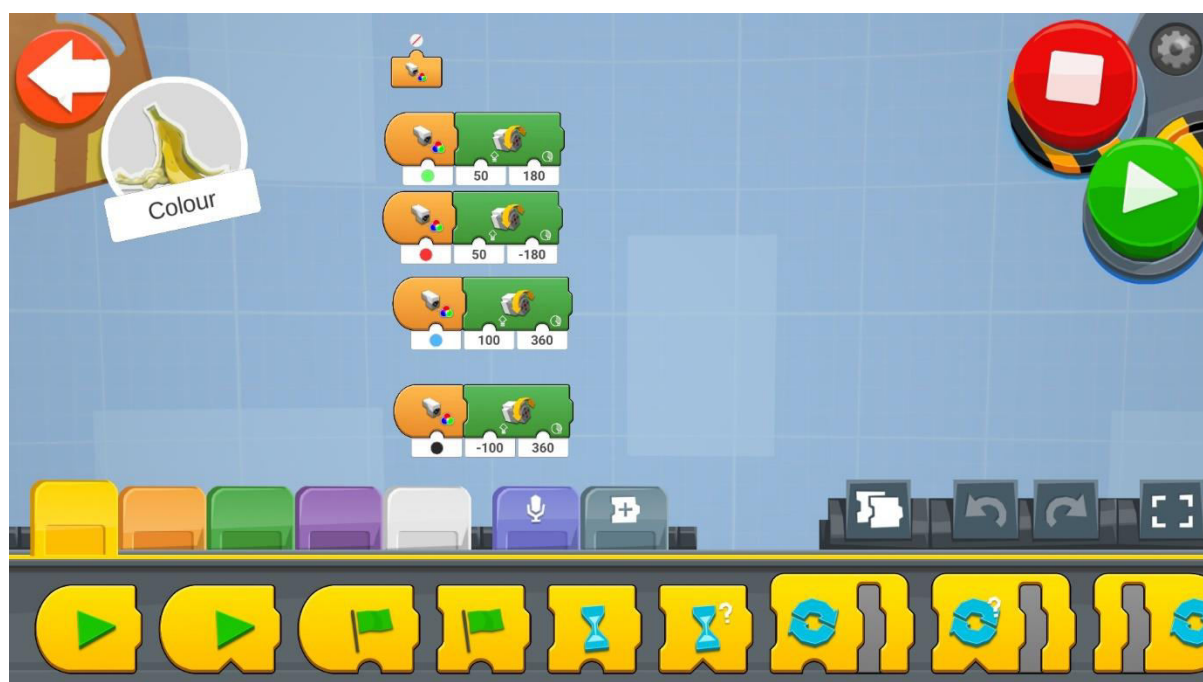
Wait for Color (*čakanie na farbu*) - Čaká na to, aby sa farba určená snímačom rovnala farbe znázornenej hodnotou pod ňou. Ak zistená farba nie je rovná zadanej hodnote, program zostane pozastavený a začne sa nasledujúca postupnosť pokynov, kým nebude splnená podmienka. Môže sa použiť sedem hodnôt: žiadna farba, čierna, modrá, zelená, žltá, červená a biela.



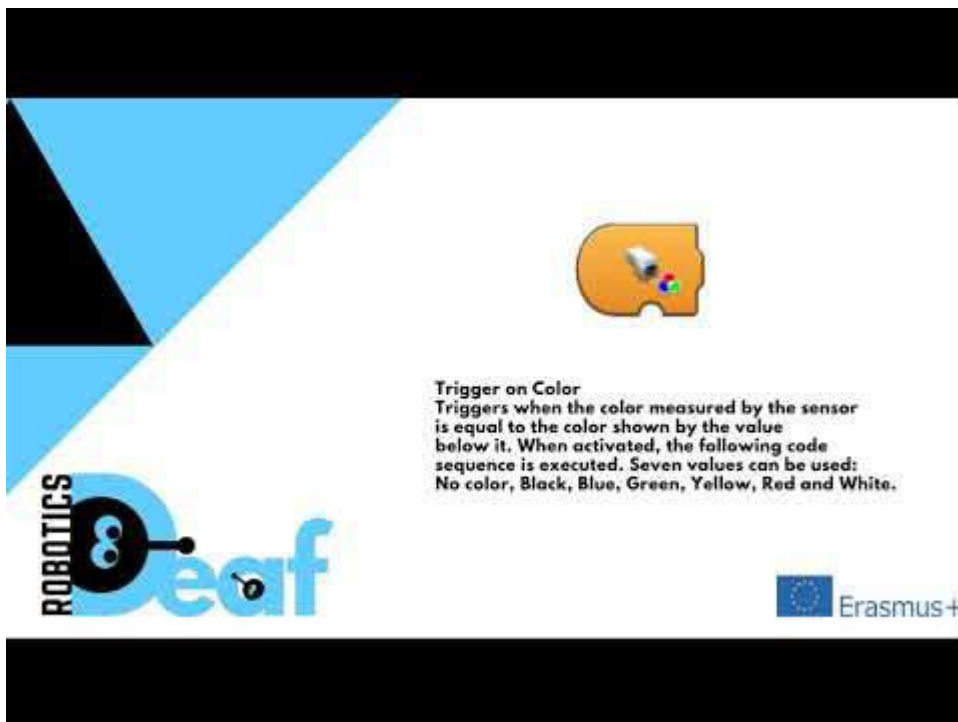
Sensor Color Reporter (*reportér snímača farieb*) - Zobrazuje aktuálnu farbu nameranú snímačom v reálnom čase. Aby sa dal použiť v programe, musí byť umiestnený na spodnej časti ostatných tehál. Uvidíte sedem hodnôt: žiadna farba, čierna, modrá, zelená, fialová, červená a biela.

Ukázkové programy na zisťovanie farieb

Predtým, ako uvidíme programy na detekciu farieb, najskôr si musíme nainštalovať interaktívny motor.



1. Prvý blok iba indikuje v reálnom čase farbu objektu pred snímačom.
2. Druhý program sa aktivuje, keď je objekt zelenej farby pred senzorom spoločnosti Smarc. Spoločnosť Smarc potom pohne vrtuľou pri rýchlosti 50 o 180 stupňov doprava.
3. Tretí program sa aktivuje, keď je objekt červenej farby pred senzorom spoločnosti Smarc. Potom spoločnosť Smarc posunie zadnú vrtuľu o 180 stupňov doľava pri rýchlosti 50.
4. Štvrtý program sa aktivuje, keď je objekt modrej farby pred senzorom Smarc. Spoločnosť Smarc potom pri rýchlosti 100 posunie zadný vrtule o 360 stupňov doprava.
5. Piaty program sa aktivuje, keď je objekt zelenej čiiernej farby pred senzorom spoločnosti Smarc. Aplikácia Smarc potom posunie zadnú vrtuľu o 360 stupňov doľava pri rýchlosti 100.



D. If/Else bloky:

Príkaz If / Else (Ak / Inak) umožňuje spoločnosti Smarc ľahko sa rozhodovať na základe vstupov zo senzora. Ak je podmienka splnená (to znamená, že je PRAVDA), program Smarc vykoná konkrétny blok kódu. Ak nie je podmienka splnená (čo znamená, že je NEPRAVDA), program Smarc vykoná ďalší blok kódu..



If/Else (*Ak/inak*)- ak je podmienka True, vykonajte hornú sekvenciu, inak ak je nepravda, vykonajte spodnú sekvenciu.



Equal Operator (*rovnaký operátor*) - vráti hodnotu True, keď sa vstup zo snímača (farba / vzdialenosť / okolité svetlo) rovná hodnote.

Less Than
je vstup zo
hodnota.



Operator (*menej ako operátor*) - vráti hodnotu True, ak snímača (farba / vzdialenosť / okolité svetlo) menší ako



Greater Than Operator (*viac ako operátor*) - vráti hodnotu True, ak je vstup zo snímača (farba / vzdialenosť / okolité svetlo) väčší ako hodnota.



Not Equal Operator (*nie je rovnocenný operátor*) - vráti hodnotu True, ak sa vstup zo snímača (farba / vzdialenosť / okolité svetlo) nerovná hodnote

Meranie množstva odrazu svetla: Bloky odrazu meracieho svetla:



Sensor Light Level Reporter (*reportér úrovne snímača osvetlenia*) – V reálnom čase zobrazuje aktuálnu hladinu okolitého svetla nameranú snímačom. Aby bolo možné ho použiť v programe, musí byť pripojený k spodnej časti ostatných blokov. Môže označovať hodnoty od jednej (1) do desať (10). Jeden je najtmavší a desať najjasnejších.

II. Nasledujúce múry so Smarc

A. Nasledujúce múry

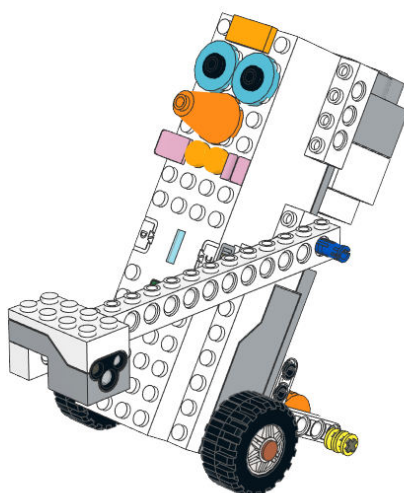
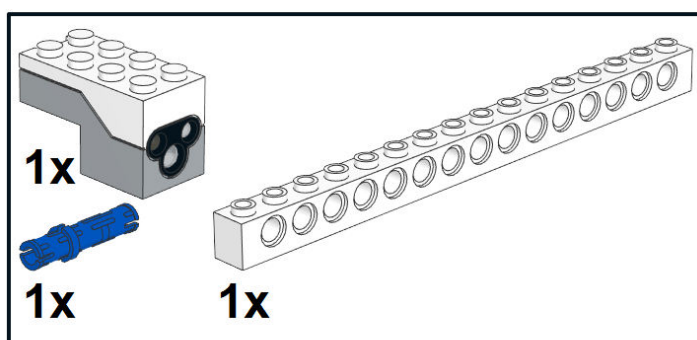
V tejto časti uvidíme, ako môže Smarc cestovať v oblasti sledovaním jednej strany steny alebo jednej strany objektu.

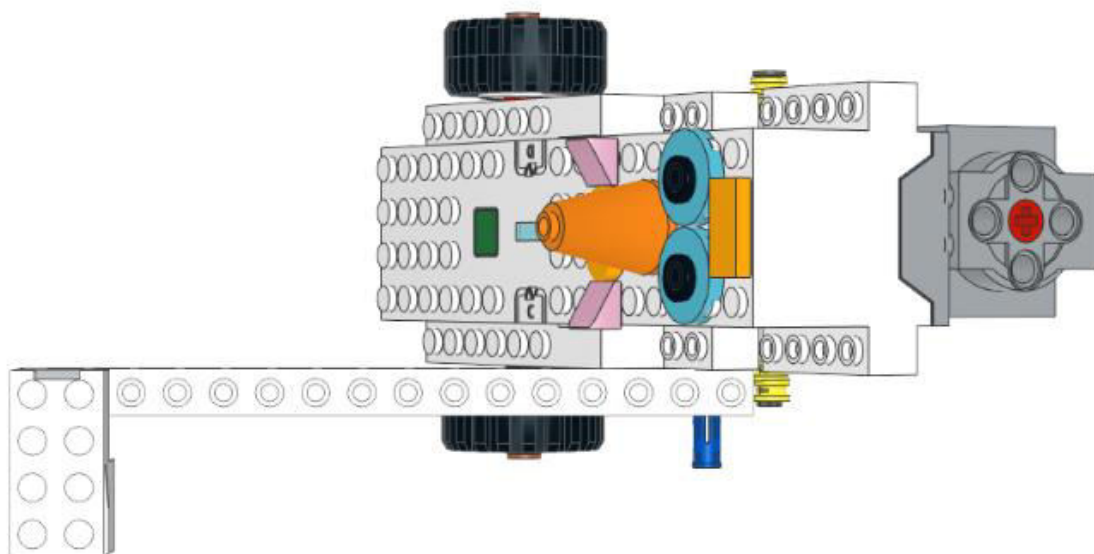
Tieto programy, ktoré reagujú na „vstupy“ získané z okolitého prostredia, sa nazývajú systémy „Spätná väzba“. Volajú sa takto, pretože prostredníctvom senzorov monitorujú zmeny alebo situácie a reagujú zmenami - „výstupom“ v systéme a v našom prípade je systémom náš robot - Smarc.

Budovanie senzora

Aby mohol Smarc cestovať v konštantnej vzdialenosti od steny, vstupom, ktorý sa použije, je načítanie zo senzora vzdialenosti. Výstupom budú motory spoločnosti Smarc, ktoré upravujú smer jazdy.

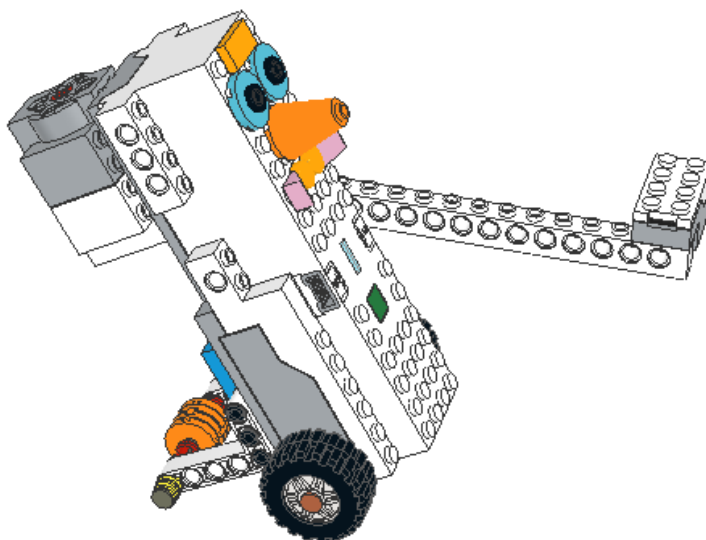
Prvá vec, ktorú budeme musieť urobiť, bude pripevniť rozšírenie v prednej časti Smarca, kam umiestnime farebný a diaľkový senzor.





Potom, čo sme pripravili Smarc so senzorom vzdialenosti, budeme musieť vytvoriť dráhu so stenami, kam sa bude Smarc pohybovať.





Pri vytváraní stopy nezabudnite, že presná strana steny, ktorá sa bude sledovať, bude smer, ktorým sa snímač pozerá. Na nastavenie trasy môžete použiť akékoľvek predmety, ktoré nájdete v učebni. Môžu to byť krabice alebo knihy, ale len sa uistite, že povrch, ktorý monitorujú snímače, je rovný.

Ukázkové programy

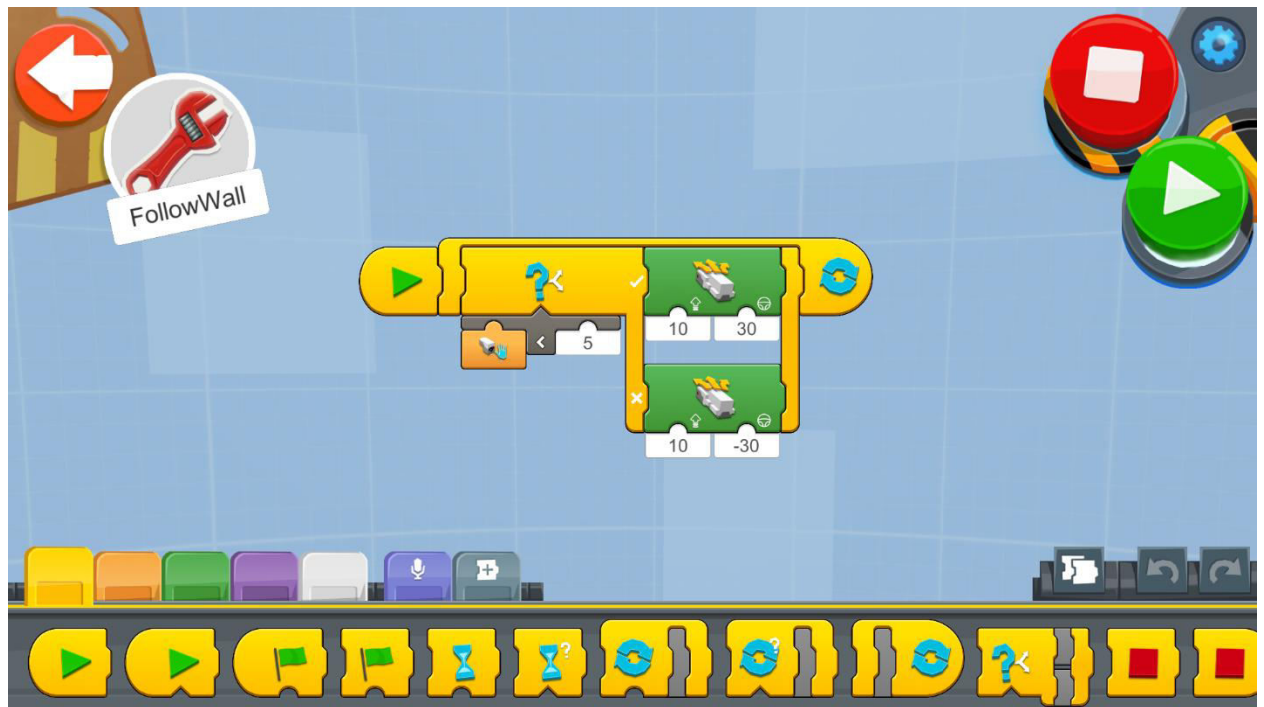


Foto 18- Vyvarovanie sa prekážkam

C Vytvorte nový projekt na Creative Canvas a spustíte blok kódu (pre aplikácie pre Android a iOS nájdete bloky na Creative Canvas of Vernie)

1. **Infinity Loop (nekonečná slučka)** spustí kód opakovane, aby Smarc neustále kontroloval objekt na ľavej strane a podľa toho konal.
2. Vo vnútri slučky sa nachádza blok **If / Else (ak/inak)**, ktorý porovnáva vstup snímača, a ak je výsledok porovnania pravdivý, dá pokyn spoločnosti Smarc, aby sa podľa toho pohla.
3. V bloku **If / Else (ak/inak)**, sa nachádza **Less Than Operator (operátor menej ako)** ktorý vráti true, keď je vstup zo senzora, v tomto prípade je to vzdialenosť, je menšia ako hodnota 5 a false, ak je rovnaký alebo väčší.
4. Ak je výsledok podmienky **If / Else True (ak/inak pravda)**, potom sa Smarc posunie o 30 stupňov doprava pomocou **Drivebase Move Steering block (bloku riadenia pohonu Drivebase)** a ak je podmienka False, potom sa Smarc posunie o 30 stupňov doľava.



An Infinite Loop will run the code repeatedly so that Smarc checks all the time for an object on its left-hand side and acts accordingly.

ROBOTICS 8 **Deaf**

 Erasmus+

The image shows a Scratch code block with an 'Infinite Loop' icon (a circular arrow) and a black arrow pointing to it. The background of the slide features a blue and black geometric design on the left and the 'ROBOTICS 8 Deaf' logo. The Erasmus+ logo is in the bottom right corner.

steny - <https://youtu.be/YzPEg9krcjA>

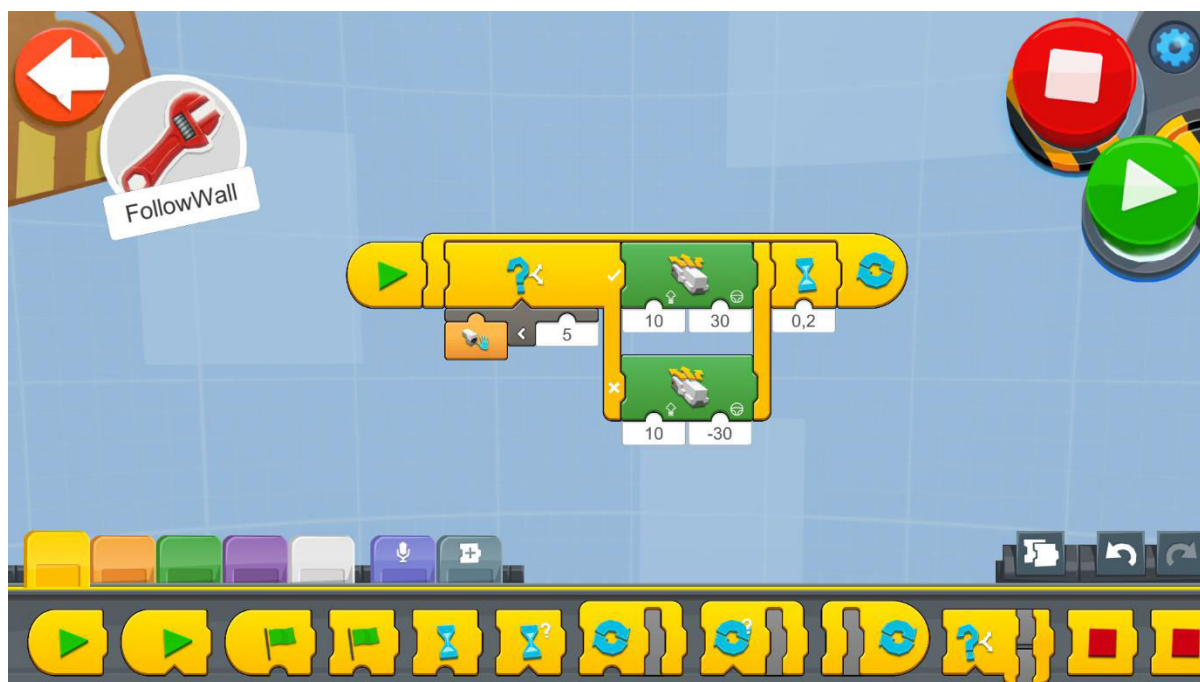


Foto 19—Vyvarovanie sa prekážkam

Poznámka pre učiteľa: Môžeme pridať inštrukciu Wait for Time block (**čakanie na časový blok**), aby sme pozastavili vykonávanie o 0,2 sekundy, aby sme spomalili rýchlosť spracovania ďalšieho čítania zo senzora.

Táto pauza bude viesť robot po dobu 0,2 sekundy pred kontrolou vzdialenosti od objektu. Bez oneskorenia 0,2 sekundy by slučka bežala maximálnym tempom a v niektorých situáciách by zariadenie Move Hub dostávalo príkazy natoľko, že by Smarc nebol schopný odpovedať a nútil ho, aby sa pohyboval náhodne.

..

III. Nasledujúce riadky so Smarcom

A. Chodte po čiare

V tejto časti uvidíme, ako môže Smarc cestovať po vopred určenej ceste označenej čiernou čiarou na bielom pozadí. Táto úloha sa nazýva sledovanie riadkov a je jednou z najbežnejších úloh v robotike. Existuje tiež veľa súťaží, ktoré zahŕňajú túto úlohu a trate sa môžu zložiť z líní, pričom trate majú ostré zákruty a križovatky!

Tento typ pohybu robotov sa v mnohých prípadoch používa aj komerčne. Jedným príkladom môžu byť sklady spoločnosti Amazon, kde sa výrobky musia zhromažďovať z rôznych miest a zhromažďovať v krabici, aby sa dali ľahko zabaliť a odoslať. Lego podobný systém používa aj pri výrobe nových dielov vo svojej továrni. Roboty sledujúce čiaru môžu sledovať zreteľne vyznačenú čiaru na podlahe alebo môžu sledovať kovové drôty, ktoré detekujú pomocou magnetických snímačov.

Smarc bude nasledovať čiaru na trati. Aby sa tak stalo, musí byť LEGO BOOST Color and Distance sensor (*LEGO BOOST senzor farieb a vzdialenosti*) na podlahe otočený smerom nadol. Čiara a pozadie musia byť farby s vysokým kontrastom. Môžeme mať napríklad čiernu čiaru s bielym pozadím alebo bielu čiaru s čiernym pozadím.

Ďalšia vec, ktorú budeme musieť urobiť, je vytvoriť trať, kam sa bude pohybovať Smarc.

Jedným jednoduchým spôsobom, ako vytvoriť cesty, je pripevniť čiernu pásku na svetlú podlahu alebo vytlačiť hrubé čierne cesty na biely papier.

Tu je jeden príklad jednoduchého sledovania trate. Tento môžete použiť na to, aby si Smarc precvičil alebo vytvoril svoj vlastný.

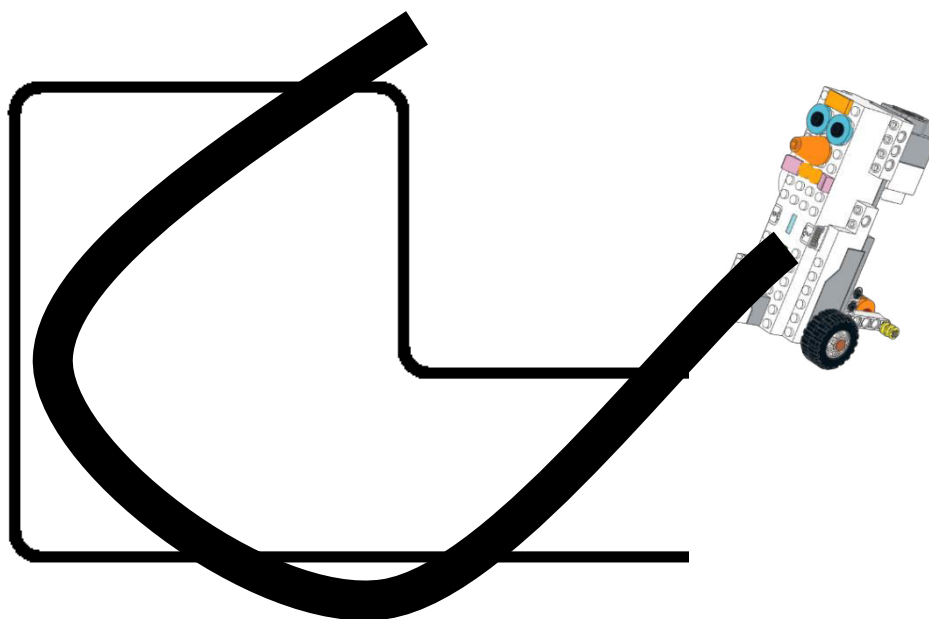


Foto 20- Chodte po línii

V ideálnom prípade by sa Smarc mal zamerať na to, aby bol snímač medzi čiernou čiarou a bielym pozadím, ako je to znázornené na obrázku nižšie. Nechceme, aby bol snímač iba na čiernej čiare alebo iba na bielom pozadí.

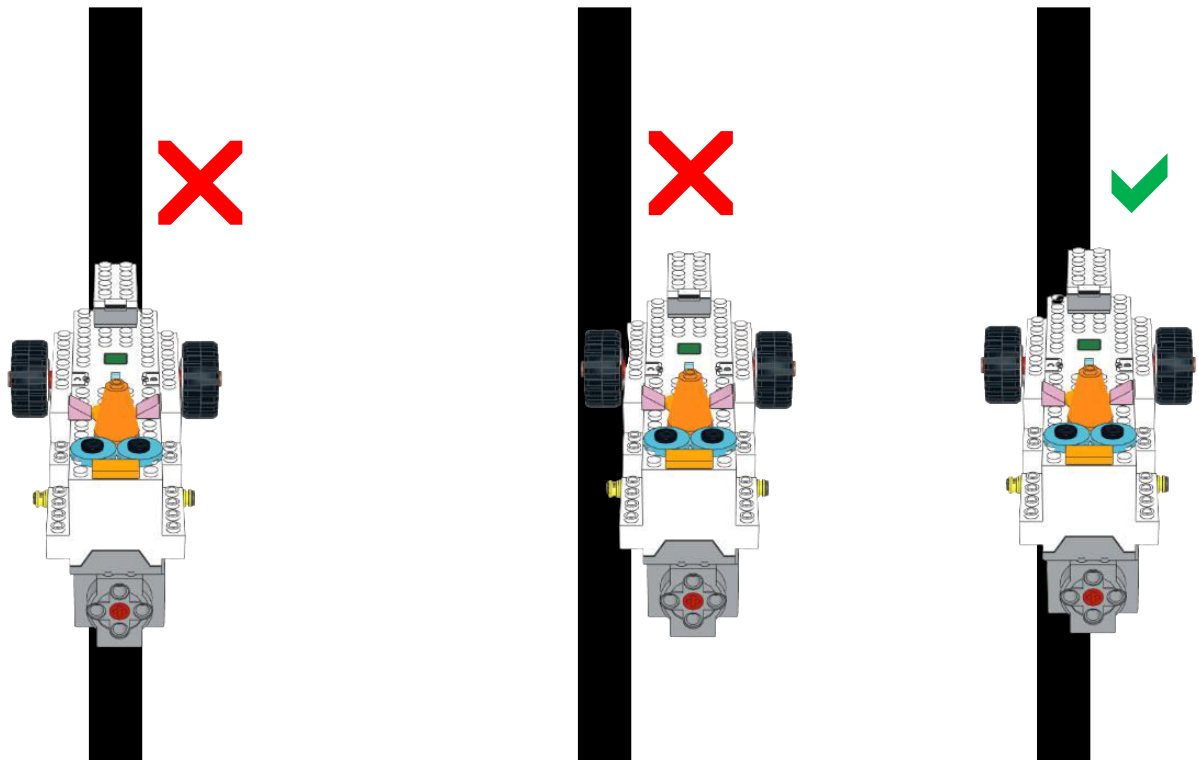


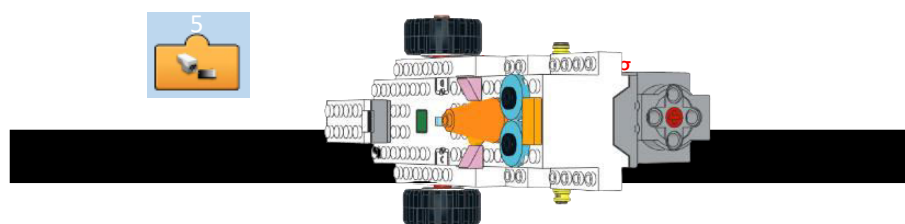
Foto 21- chodte po línii - Smarcova pozícia

Potom budeme musieť vykonať niekoľko meraní zo senzora pre našu stopu, ktorú sme vytvorili predtým v triede.

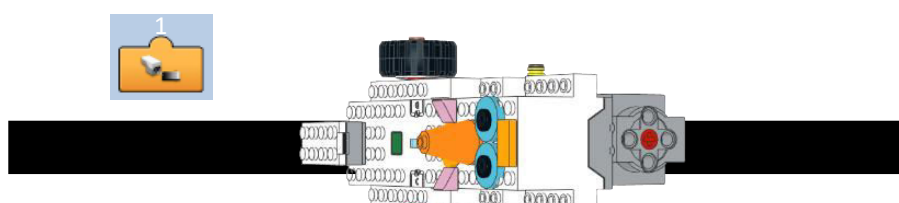
Aby sme tak mohli urobiť, budeme musieť použiť blok Sensor Light Level Reporter (*senzora snímača úrovne osvetlenia*).

Existujú dva spôsoby, ako môžeme vykonať merania:

- a. Vykonaním jedného merania, keď je snímač umiestnený medzi čiernou čiarou a bielym pozadím, ako je to znázornené na obrázku.

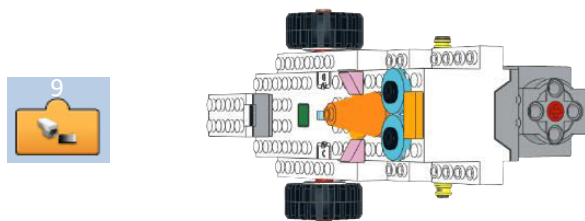


- b. b. Pri umiestnení snímača vykonáte dve merania:
 - i) iba na čiernu čiaru



Nasledovať podľa čiary - urobte 2 merania na čiernej čiare

ii) iba na bielom pozadí



A

potom nájdite priemer z dvoch vykonaných meraní:

$$\frac{(\text{meranie čiernej čiary} + \text{meranie bieleho pozadia})}{2} = \text{Požadované čítanie, aby ste zostali na správnej ceste}$$

Príklad výpočtu:

$$\frac{(1+9)}{2}=5$$

Poznámky:

Pri vykonávaní týchto odpočtov by sa mali brať do úvahy niektoré vonkajšie faktory, ktoré ovplyvňujú odpočty senzorov. Môže to byť typ papiera použitého na skladbu (snažte sa vyhnúť lesklému papieru), tieň rôznych predmetov, ako sú steny, knihy, alebo dokonca tieň študentov odrážajúce sa na skladbe. Skúste svoju stopu umiestniť mimo slnečného svetla okna svojej triedy a umiestnite ju priamo pod LED lampu, najlepšie preto, lebo sa svetlo šíri rovnomernejšie. Spravidla by ste sa mali snažiť byť dôslední a umiestniť Smarca a stopu po línii do rovnakej presnej oblasti, ako je to možné.

Ďalším dôležitým problémom je farba čiary a pozadia. Môžu existovať rôzne tóny čiernej a bielej farby, takže vyššie uvedené merania sú vhodné pre konkrétny prípad. Mali by ste si urobiť vlastné merania, ktoré budú vhodné pre vašu konkrétnu stopu, ktorú ste vytvorili.

Ukážkové programy na sledovanie tejto línie

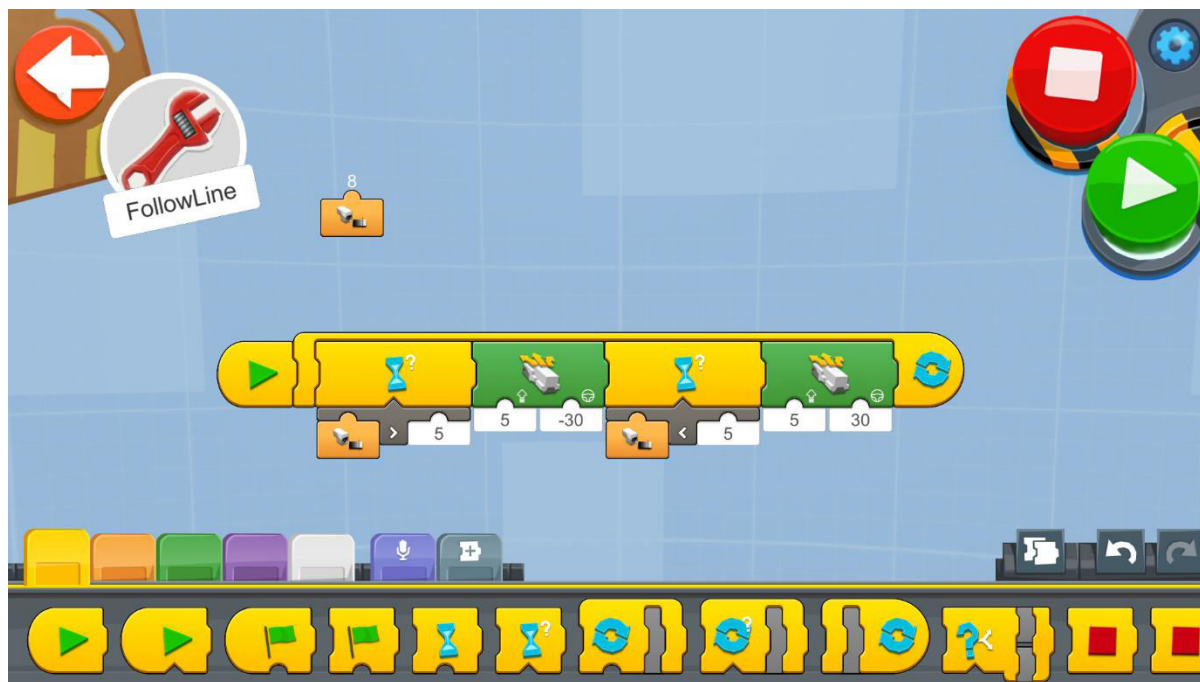


Foto 22-Program – sledovanie čiary

Vytvorte nový projekt na kreatívnom plátne a spustíte blok kódu:

1. Infinite Loop (*nekonečná slučka*) spustí kód opakovane, takže Smarc neustále kontroluje načítanie zo senzora a podľa toho koná.
2. Vo vnútri slučky je Wait for True Block (*čakanie na skutočný blok*), ktorý čaká na splnenie podmienky, aby mohol prejsť na ďalší blok inštrukcií.
3. V sekcii Wait for True Block (*čakanie na skutočný blok*) je operátor Greater Than Operator (*viac ako operátor*), ktorý vráti True, keď je vstup zo senzora, v tomto prípade senzora vzdialenosti a farby, väčší ako hodnota 5 a false, ak je rovnaký alebo menší. To sa vráti do platnosti, ak sa Smarc vzdďaľuje od čiernej čiary a snímač je otočený iba k bielemu pozadiu.
4. Ak je výsledok podmienky Wait for True Block (*čakania na skutočný blok*) pravdivý, potom sa Smarc posunie o 30 stupňov doľava (pokúsi sa priblížiť k čiernej čiare) pomocou bloku **riadenia Drivebase Move Steering** rýchlosťou 5 (snažíme sa udržiavať nízku rýchlosť).
5. V druhom Wait for True Block (*čakanie na skutočný blok*) je Less Than Operator (*menej ako operátor*), ktorý vráti True, keď je vstup zo senzora, v tomto prípade senzora vzdialenosti a farby, menší ako hodnota 5 a false, ak je rovnaký alebo väčší. To sa vráti do platnosti, ak sa Smarc priblíži k čiernej čiare a snímač bude smerovať iba k čiernej farbe čiary.
6. Ak je výsledok druhej podmienky Wait for True Block (*čakanie na skutočný blok*), potom sa Smarc posunie o 30 stupňov doprava (pokúsi sa odkloniť od čiernej čiary) pomocou **bloku Drivebase Move Steering** rýchlosťou 5.
7. Nakoniec je vždy dobré mať v reálnom čase údaje o hodnotách, ktoré senzor odosiela do aplikácie. V tomto prípade sa použije blok Sensor Light Level Reporter (*reportér úrovne svetla*).

Poznámka: Tento program bude fungovať dobre, iba ak je snímač umiestnený na pravej strane linky, a nie na ľavej strane!

Robotics4Deaf - Modul1. Sledovanie podľa čiary- <https://youtu.be/AX0kJ5ngTeY>

Druhý nasleduje líniový program:

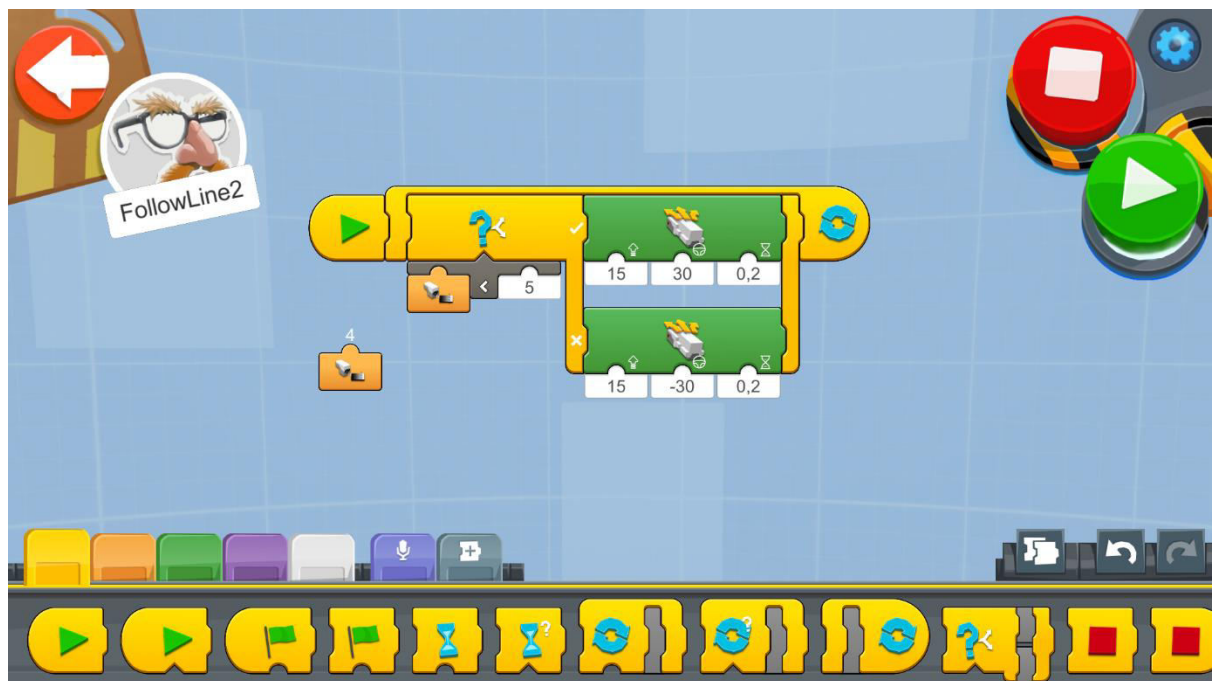


Foto 23- Program 2 – Sledovanie čiary

Vytvorte nový projekt na kreatívnom plátne a spustíte blok kódu:

1. Infinite Loop (*nekonečná slučka*) spustí kód opakovane, takže Smarc neustále kontroluje načítanie zo senzora a podľa toho koná.
2. Vo vnútri slučky je blok If / Else (*ak/inak*), ktorý, ak má hodnotu PRAVDA, sa posunie Smarc doprava (pokúsi sa posunúť od čiernej čiary) a ak má hodnotu FALSE, Smarc sa pohne doľava (pokúsi sa pohnúť bližšie k čiernej čiare).
3. Blok Compare Less Than (*porovnať menej ako*) nastaví podmienku pre blok Prepínač. Blok Compare Less Than (*porovnať menej ako*) porovnáva výstup bloku Sensor Light Level Reporter (*reportéra snímača úrovne svetla*) s hodnotou 5. Vráti hodnotu true, keď je snímač bližšie k čiernej čiare, a hodnotu false, keď sa vzdiali.
4. Ak je výstup bloku If / Else (*ak/inak*) pravdivý, potom blok Drivebase Move Steering for Duration (*DriveBase posunu riadenia na dobu trvania*) posunie Smarc rýchlosťou 15 v uhle 30 stupňov doprava a bude to robiť ešte 0,2 sekundy. Ak je nepravdivé, druhý blok Drivebase Move Steering for Duration (*DriveBase posunu riadenia na dobu trvania*) bude posúvať Smarc rýchlosťou 15 v uhle 30 stupňov doľava a bude to robiť ešte 0,2 sekundy..

Poznámka:

Dôvod na pridanie trvania 0,2 sekundy je spôsobený oneskorením od času medzi čítaním odoslaným do Smarcu, kým sa Smarc rozhodne, kam sa má pohnúť. Preto keď sa Smarc posunul blízko k čiernej čiare, rozhodnutie vzdialiť sa bude musieť kompenzovať stratený čas vykonaním výpočtov a prenosom inštrukcie, kým sa Smarc stále pohyboval smerom k čiernej čiare.

Sledovanie podľa čiary 2 - <https://youtu.be/BzJ5nq2EHLU>

IV. Detekcia zvuku pomocou protokolu Smarc

A. Reakcia na zvuk

Zvukom aktivovaný robot

V tejto časti uvidíme, ako dokážeme ovládať Smarc zvukmi, ako je napríklad tlieskanie rukami, a použitím rôznych intenzít zvuku pomocou blokov zvukového senzora.

V prípade Smarc je zvukovým senzorom vlastne zabudovaný mikrofón, ktorý sa nenachádza v Move Hub. Namiesto toho sa spolieha na mikrofón, ktorý je umiestnený vo vnútri zariadenia, ku ktorému je Smarc pripojený, napríklad k tabletu, notebooku alebo smartfónu.

Aplikácia Lego Boost rozpozná, aký hlasný zvuk je. Preto ho môžeme naprogramovať tak, aby reagoval rôzne v závislosti od prijatej intenzity zvuku.

B. Bloky zvukového senzora:



Trigger on Sound Level (*spúšťač na úrovni zvuku*) - Spustí sa, keď je hladina zvuku nameraná snímačom vyššia ako hladina zvuku znázornená hodnotou pod ním. Po spustení vykoná nasledujúcu sekvenciu kódov. Bude mať 11 hodnôt od 0 do 10.



Wait for Sound Level (*čakanie na úroveň zvuku*)- Čaká, kým intenzita zvuku nameraná snímačom nebude vyššia ako hladina zvuku uvedená nižšie. Ak zaregistrovaná hladina zvuku nie je vyššia ako zadaná hodnota, program

zostane pozastavený a začne sa nasledujúca séria pokynov, kým nebude splnené dané kritérium. Vezmete jedenásť hodnôt od 0 do 10.



Sound Level Reporter (*reportér úrovne zvuku*) - Zobrazuje najnovšiu zvukovú frekvenciu vypočítanú snímačom v reálnom čase. Aby sa dal použiť v programe, musí byť pripojený k spodnej časti ostatných tehál. Bude predstavovať hodnoty od 0 do 10 vrátane jedného desatinného čísla, napr. 7,8.

Ukážkové programy na detekciu zvuku

Program 1

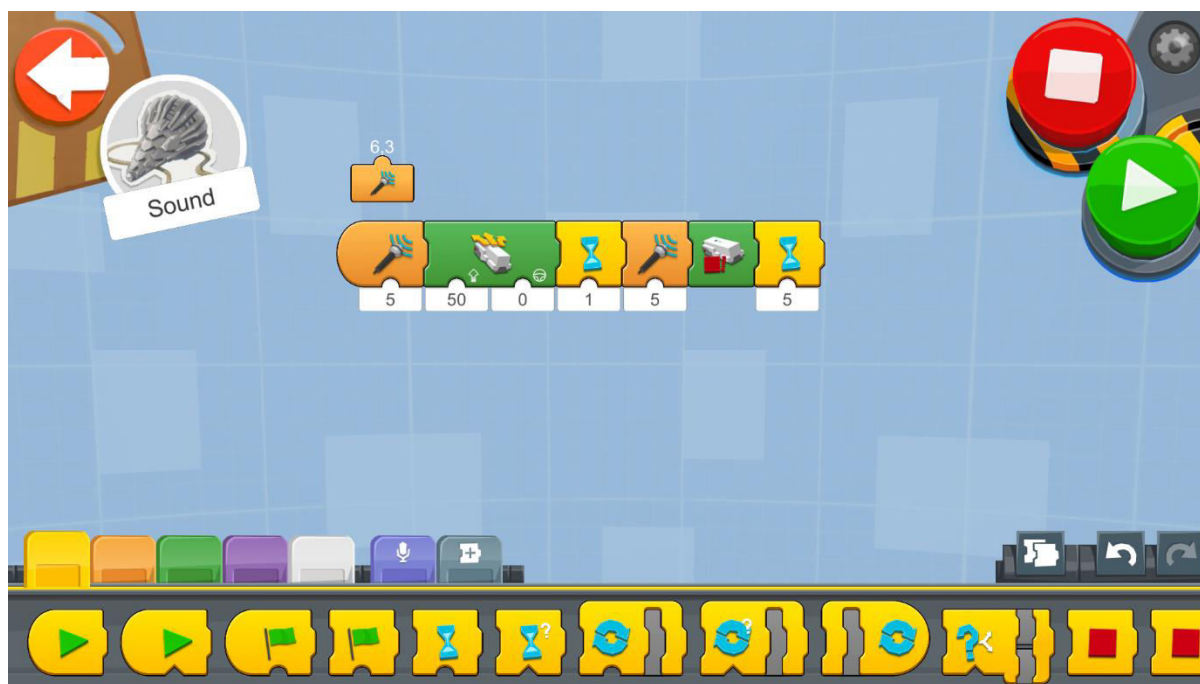


Foto 24-Program 1- Detekcia zvuku

Vytvorte nový projekt na kreatívnom plátne a spustite blok kódu:

1. Kód sa spustí, keď sa spustí blok Trigger on Sound Level (*spúšťač na úrovni zvuku*).
Stane sa to, keď sa zistí šum na úrovni vyššej ako 5.
2. Ak je detekovaný zvuk, Smarc sa pohybuje dopredu pomocou bloku Drivebase Move Steering (*Drivebase posunu riadenia*) rýchlosťou 50.

3. Blok Wait for Time (*čakanie na čas*) je nastavený na 1 sekundu. Dôvodom tohto bloku je, aby sa Smarc nezmiešal s rovnakým zvukom, aký sa používa na aktiváciu programu na posun vpred a na spustenie zvyšku kódu.
4. Blok Wait for Sound Level (*čakanie na úroveň zvuku*) čaká na šum na úrovni vyššej ako 5, aby spustil vykonanie zvyšku kódu.
5. Nakoniec, keď sa aktivuje blok Wait for Sound (*čakanie na zvuk*), aktivuje sa Drivebase Stop a zastaví Smarc v pohybe.



Detekcia zvuku 1 - <https://youtu.be/3xrI2SGrlh8>

Program 2

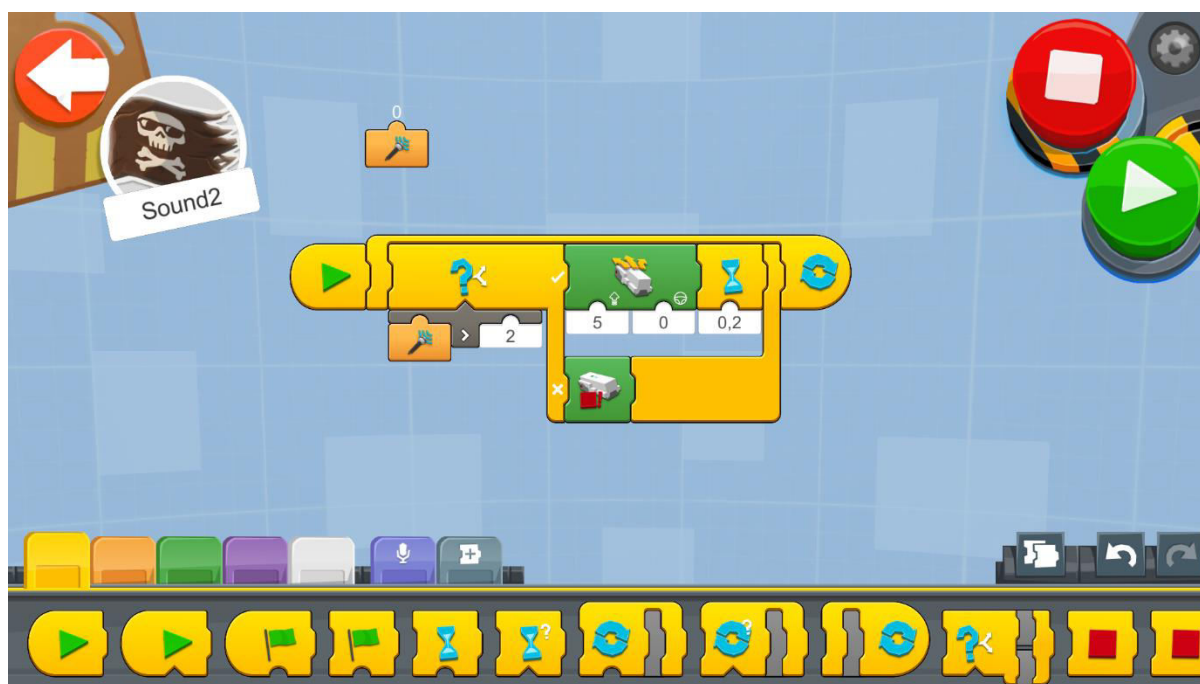


Foto 25- Program 2 – Detekcia zvuku

Vytvorte nový projekt na kreatívnom plátne a spustíte blok kódu:

1. Infinite Loop (*nekonečná slučka*) bude spúšťať kód opakovane, aby Smarc neustále kontroloval čítanie zo senzora a podľa toho konal.
2. Vo vnútri slučky sa nachádza blok If / Else (*ak/inak*), ktorý, ak má hodnotu TRUE, sa Smarc pohne dopredu a ak je FALSE, Smarc sa zastaví.
3. Blok Compare Greater Than (*porovnať väčšie ako*) nastavuje podmienky pre blok Prepínač. Blok Compare Greater Than (*porovnať väčšie ako*) porovnáva výstup bloku Sensor Sound Level Reporter (*reportéra snímača úrovne zvuku*) s hodnotou 2. Vráti hodnotu true, keď snímač zistí hladinu zvuku vyššiu ako 2, a hodnotu false, ak je rovný alebo menší ako 2.
4. Ak je výstup bloku If / Else (*ak/inak*) pravdivý, blok Drivebase Move Steering (*Drivebase riadenia pohonu*) presunie Smarc rýchlosťou 55, v uhle 0 stupňov, a bude to robiť ešte 0,2 sekundy. Ak je nepravdivé, blok Drivebase Stop zastaví Smarc v pohybe.



https://youtu.be/oMq_olyajE0

V. Navigácia v Smarc pomocou diaľkového ovládača

A. Diaľkové ovládanie

V tejto časti uvidíme, ako sa dokáže Smarc pohybovať pomocou diaľkového ovládača.

Existuje mnoho rôznych spôsobov komunikácie s robotom. Roboty sa najčastejšie ovládajú pomocou káblového alebo bezdrôtového ovládača alebo môžu bežať autonómne získavaním pokynov od svojho riadiaceho programu (čo sme robili so programom Smarc v predchádzajúcich kapitolách).

Diaľkovo ovládané roboty môžu mať rôzne vedecké použitia vrátane nebezpečných prostredí, ako sú extrémne horúčavy alebo chladu a rádioaktívne miesta, práce v hlbokom oceáne na umiestňovanie potrubí a káblov, ako aj prieskum vesmíru a planét.

B. Bloky diaľkového ovládania:



Joystick Widget Show (zobrazenie miniaplikácie Joystick) - Keď je tento blok aktivovaný, predstavuje miniaplikácia Joystick v aplikácii Lego BOOST.



Joystick Widget Hide (*skrytá miniaplikácia Joystick*) - Keď je tento blok aktivovaný, skryje miniaplikácia Joystick z aplikácie Lego BOOST.



Joystick Widget Speed Reporter (*rýchlosť joysticku – reporter*) - Zobrazí v reálnom čase aktuálnu rýchlosť Joystick Widget (-100..100). Aby bolo možné ho použiť v programe, musí byť pripojený k spodnej časti ostatných blokov. Vo väčšine prípadov sa používa ako vstup riadenia do bloku riadenia Drivebase Move. Môže zobrazovať hodnoty od -100 do 100 vrátane dvoch desatinných hodnôt, napríklad 78,89.



Joystick Widget Steering Reporter (*reportér riadenia joysticku*) - Zobrazuje aktuálne riadenie Joystick Widget v reálnom čase (-100..100). Aby bolo možné ho použiť v programe, musí byť pripojený k spodnej časti ostatných blokov. Vo väčšine prípadov sa používa ako vstup rýchlosti do bloku riadenia Drivebase Move. Môže zobrazovať hodnoty od -100 do 100 vrátane dvoch desatinných hodnôt, napríklad 78,89.

Ukážkové programy pre diaľkové ovládanie

Program 1



Foto 26-Program – Diaľkové ovládanie

Vytvorte nový projekt na kreatívnom plátne a spustíte blok kódu:

1. Na Joystick Widget Show (miniaplikácii Joystick) sa zobrazí ovládač joysticku v aplikácii (pozri obrázok).
2. Infinite Loop (nekonečná slučka) spustí kód opakovane, takže Smarc po celú dobu kontroluje čítanie z joysticku a koná podľa toho.
3. Vo vnútri slučky sa nachádza blok Drivebase Move Steering (Drivebase riadenia pohonu), ktorý bude brať ako rýchlosť načítanie čítania bloku Joystick Widget Speed Reporter (miniaplikácie rýchlosti joysticku – reportér) a ako smerovanie načítanie bloku Joystick Widget Steering Reporter (miniaplikácie riadenia Joysticku)

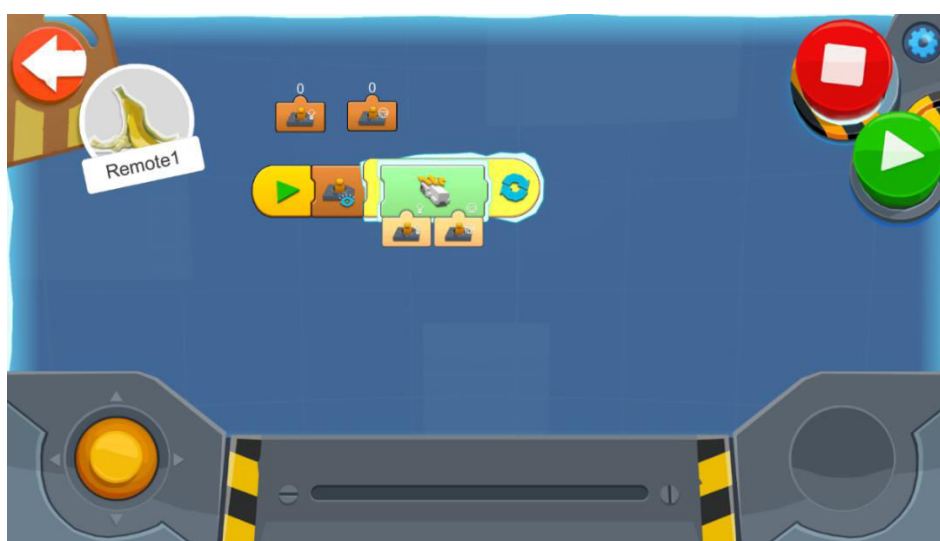


Photo 27-Program - Joystick Controller

Program 2



Foto 28- Program 2 – Diaľkové ovládanie

Niekedy potrebujeme presnejšiu kontrolu nad Smarcom a vstupná rýchlosť je na to príliš vysoká. Na tento účel môžeme urobiť trik, aby sme znížili vstupnú rýchlosť, ktorú odosiela blok Joystick Widget Speed Reporter (*miniaplikácie reportéra rýchlosti joysticku*). Na zadanie rýchlosti bloku Drivebase Move (*pohon*) sa používa operátor divízie. Vstup bloku Joystick Widget Speed Reporter (*miniaplikácie reportéra rýchlosti joysticku*) sa najskôr vydělí 4 a potom sa hodnota odošle ako hodnota rýchlosti. Napríklad, ak je hodnota načítaná z joysticku 100, rýchlosť bude:

$$100 \div 4 = 25$$

Skutočná rýchlosť by teda bola 25..



VI. Používanie premenných so Smarcom

A. Matematika a výpočty

V tejto časti uvidíme, ako je možné použiť rôzne matematické operácie v kombinácii s premennými.

Čo sú to však premenné?

Kód, ktorý sme doteraz vytvorili, si nemusel pripomínať žiadne hodnoty, pretože čísla, ktoré sme použili, napríklad odrazené svetlo a frekvencia zvuku, pochádzajú zo snímača.

Existujú prípady, keď budeme chcieť, aby si Smarc uložil a zapamätal hodnotu, ako napríklad číslo, aby sme ju mohli neskôr použiť vo svojom programe.

Aby ste mohli ukladať čísla do pamäte spoločnosti Smarc, musíte každému miestu v pamäti najskôr priradiť názov, aby ste predišli zámene.

V príklade nižšie máme dve pamäťové miesta - kontajnery.

Prvý kontajner má štyri bloky Red Lego, takže môžeme povedať, že pamäťovému miestu „RED-ČERVENÉ“ je priradené číslo 4. RED = 4.

Druhý kontajner má tri bloky bieleho lega, takže môžeme povedať, že pamäťové miesto „WHITE-BIELE“ má priradené číslo 3. WHITE = 3.

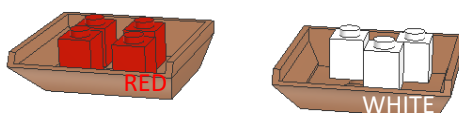


Foto 29-Červené a biele Lego Boost

Aplikácia Lego Boost vám dáva možnosť pomenovať pamäť pamäte pomocou iba jedného písmena anglickej abecedy (obrázok) alebo symbolu inej symbolickej abecedy (obrázok).

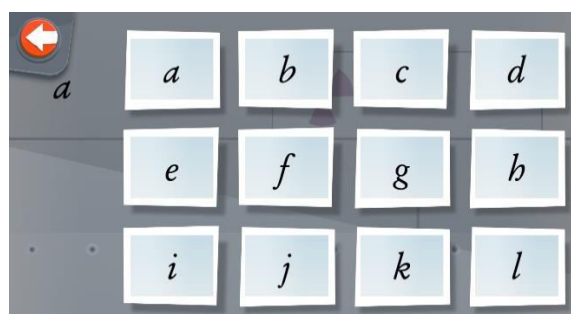


Foto 30- Anglická abeceda

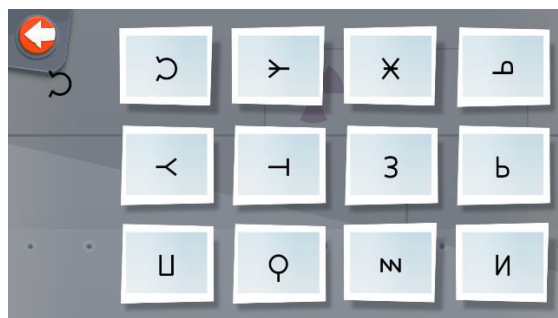


Foto 31-Symbolická abeceda

B. Bloky operátora:



Addition Operator (*operátor sčítania*) - Vráti výsledok prídania čísla k inému číslu.



Subtraction Operator (*operátor odčítania*) - Vráti výsledok čísla odpočítaného od iného čísla.



Multiply Operator (*operátor znásobenia*) - Vráti výsledok čísla vynásobeného iným číslom.

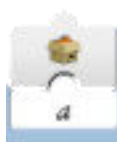


Division Operator (*operátor delenia*) - Vráti výsledok čísla vydelený iným číslom.



Equal Operator (*rovnaký operátor*) - Vráti hodnotu True, ak sa číslo rovná hodnote.

C. Variabilné bloky:



Variable Read Local (*variabilné lokálne čítanie*) - Zobrazí v reálnom čase číslo uložené v lokálnej premennej.



Variable Write Local (*premenný lokálny zápis*) - Aktualizuje miestnu premennú tak, aby ukladala uvedené číslo.

Vzorové programy pre matematiku a výpočty

Program 1

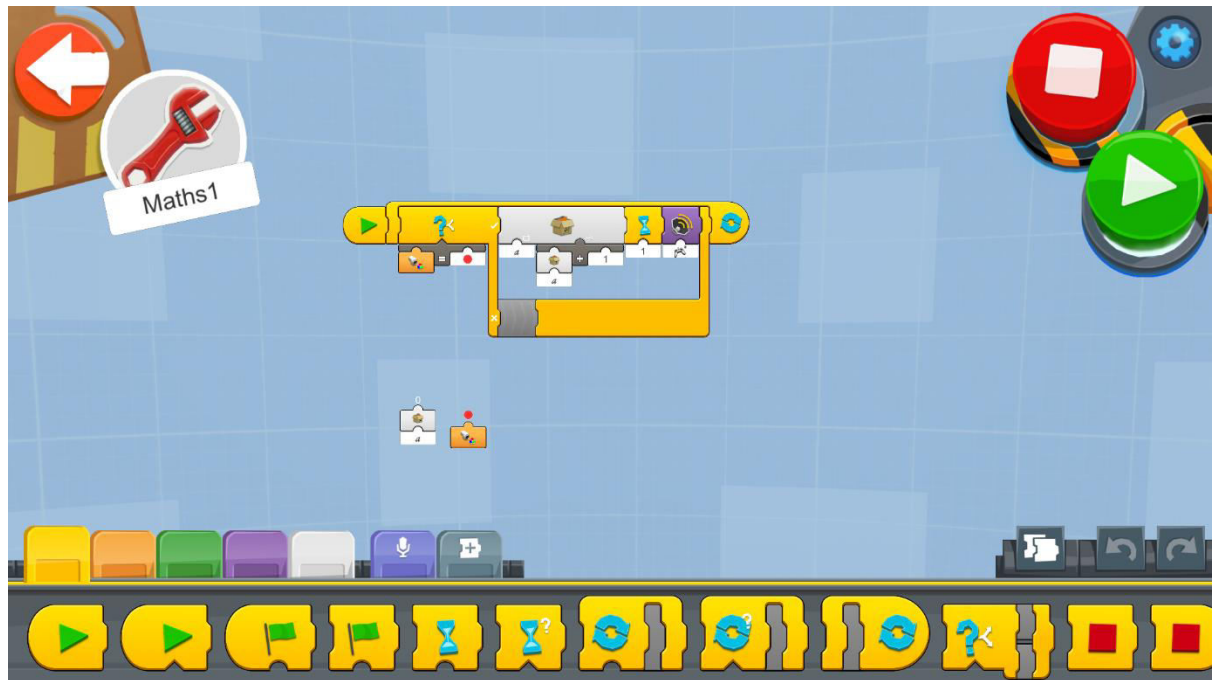


Foto 32-Program – Matematika a výpočty

Vytvorte nový projekt na kreatívnom plátne a spustite blok kódu:

1. Infinite Loop (*nekonečná slučka*) spustí kód opakovane, takže Smarc neustále kontroluje načítanie zo senzora a podľa toho koná.
2. Vo vnútri slučky je blok If / Else (ak/inak), ktorý, ak senzor detekuje RED-ČERVENÚ farbu, obsah premennej „a“ sa zvýši o 1.
3. Po každom zvýšení premennej o 1 sa blokovanie čakania aktivuje na 1 sekundu. Dôvodom je skutočnosť, že premenná sa nezistí viac ako raz, keď sa zistí červená farba.
4. Nakoniec sa Sound block triggers (zvukový blok spustí)hneď po každom zvýšení, aby používateľ vedel, že sa premenná zvýšila.

Program 2



Foto 33-Program 2 – Matematika a výpočty

Z predchádzajúceho vytvoreného programu vyplňte ďalší kód, ako je uvedené vyššie:

1. Druhý blok If / Else (*ak/inak*) vráti TRUE, ak senzor detekuje ZELENÚ farbu.
2. Ak je výsledok bloku If / Else TRUE, stredný motor sa bude otáčať rýchlosťou 50 po čas (v sekundách), ktorý sa rovná obsahu premennej „a“.



Tento dokument sa môže kopírovať, rozmnožovať alebo upravovať v súlade s vyššie uvedenými pravidlami. Okrem toho musí byť jasne uvedené potvrdenie o autoroch dokumentu a všetky príslušné časti oznámenia o autorských právach.

Všetky práva sú vyhrazené.

Copyright © in ROBOTICS4DEAF consortium, 2019-2021

