

# PREMENNÉ A MATEMATICKÉ OPERÁCIE

## ROBOT POČÍTA ZNAKY A SLEDUJE PREKÁŽKY

Tematický celok / Téma		ISCED / Odporúčaný ročník
<ul style="list-style-type: none"> <li>Softvér a hardvér – počítač a prídavné zariadenia</li> <li>Algoritmické riešenie problémov – analýza problému</li> </ul>		ZŠ 6-7.ročník – 2 vyučovacie hodiny
Požiadavky na vstupné vedomosti a zručnosti		
<ul style="list-style-type: none"> <li>poznať konštrukčné prvky LEGO EV3 zostavy</li> <li>poznať grafické programátorské prostredie LME EV3 (predchádzajúce metodiky)</li> <li>poznať základné algoritmické štruktúry: cyklus s pevným počtom opakovaní a jednoduchá podmienka</li> </ul>		
Ciele		
Žiakom osvojované vedomosti		Žiakom rozvíjané zručnosti a spôsobilosti
<p><b>Počítač a prídavné zariadenia:</b></p> <ul style="list-style-type: none"> <li>skúmať nové možnosti použitia konkrétneho hardvéru</li> <li>pracovať s prídavnými zariadeniami</li> </ul> <p><b>Analýza problému:</b></p> <ul style="list-style-type: none"> <li>realizovať návod, postup, algoritmus riešenia úlohy – interpretovať ho, krokovať riešenie, simulovať činnosť vykonávateľa,</li> <li>vyjadriť princíp fungovania návodu – objaviť a popísať vlastnými slovami princíp fungovania jednoduchého algoritmu,</li> <li>vyhľadať vzťah medzi vstupom, algoritmom a výsledkom</li> </ul> <p>Edukačná robotika - LEGO EV3 Mindstorms</p> <ul style="list-style-type: none"> <li>vytvoriť a ladiť program v grafickom programovom prostredí LEGO Mindstorms</li> <li>navrhnuť výber senzorov pre účelné ovládanie robota v konkrétnej úlohe</li> <li>používať programové bloky premenných a matematických operácií</li> </ul>		<p>Informatické myslenie:</p> <ul style="list-style-type: none"> <li>(LOG2) využitím logických zdôvodnení predpokladať správanie sa algoritmov</li> <li>(LOG3) využitím logických zdôvodnení detegovať a opravovať chyby v programoch a algoritmoch</li> <li>(ALG8) zapísať algoritmy v konkrétnom programovacom jazyku LEGO Mindstorms</li> <li>(VYH3) posúdiť správnosť postupu na základe vybraných kritérií, testovať program</li> </ul> <p>Bádateľské spôsobilosti:</p> <ul style="list-style-type: none"> <li>manipulovať s hardvérom</li> <li>diskutovať o obmedzeniach realizovaného konštrukčného postupu</li> <li>diskutovať/obhajovať o spôsobe využívania hardvérových komponentov</li> </ul>
Riešený didaktický problém		
<p>Pochopenie princípu používania premenných často predstavuje problém pre žiakov základných škôl. Pri senzorickej ovládajúcej LEGO robotov ukážeme význam využitia premenných a matematických operácií na jednoduchých príkladoch. Žiaci majú pochopiť, že senzormi zistené údaje, ktoré neskôr majú používať, musia byť uložené na jednoznačne určenom mieste.</p>		
Dominantné vyučovacie metódy a formy		Príprava učiteľa a pomôcky
<ul style="list-style-type: none"> <li>bádateľská metóda 5E</li> </ul>		<ul style="list-style-type: none"> <li>HW: Základná Lego EV3 zostava, počítač</li> </ul>



- práca vo dvojiciach s robotickou súpravou pomocou pracovného listu,
- riadený rozhovor

- SW: program EV3 Mindstorms Student Edition
- Čierna a farebná izolačná páska na vytvorenie označení čiarami
- Pracovný list pre každého žiaka

### Diagnostika splnenia vzdelávacích cieľov

Žiaci vybaví robota, ktorého postavili na predchádzajúcej hodine ďalšími senzormi a riešia úlohy.

Počas etapy vysvetľovania sa zistí, či žiaci správne pochopili funkciu a používanie premenných.

Svoje riešenia predstavia a spoločne diskutujú o využívaní hardvérových komponentov a programátorských riešení, obhajujú svoje postupy. Na konci hodiny vyplnia sebahodnotiacu kartu v pracovnom liste.

## ÚVOD

Pokračujeme v programovaní autonómnych robotov na základe senzorických zistení. Je to piata, a zároveň posledná metodika v sérii metodík pre robotiku s LEGO EV3 zostavami. Po zvládnutí ovládania motorov a niektorých senzorov teraz ukážeme význam spracovania dát získaných zo senzorických meraní a žiaci na jednoduchých príkladoch bádateľsky získajú skúsenosti s jednoduchými premennými.

V programovacom prostredí LEGO Mindstorms Education sú logicky zoskupené grafické programátorské bloky, mimoriadne vhodné pre samostatné bádanie žiakov.

Na prvých hodinách sme používali bloky zo zelených a oranžových sád (akčné a funkčné bloky), na tejto hodine využijeme bloky aj zo žltých a červených sád (senzorické a dátové operácie).

Vzorové riešenia úloh tejto metodiky nájdete v prílohe v súbore **premenne.ev3**.

Predpokladáme, že žiaci počas riešenia úloh tejto série metodík nadobudnú také skúsenosti a vedomosti, aby dokázali riešiť ďalšie vlastné projekty a prípadne aj súťažné úlohy. Po tomto základnom kurze by mali zvládnuť napríklad úlohy korešpondenčnej súťaže Robotická Liga <http://liga.robotika.sk/>. (V archíve sú námety na úlohy aj so vzorovými riešeniami.)

## OSNOVA VYUČOVACEJ HODINY:

1. **Zapojenie:** Motivačný rozhovor na tému úložiska (5 min)
2. **Skúmanie:** Počítanie čiar a prekážka (35 min)
3. **Vysvetľovanie:** Hodnotenie vytvoreného robota a programu (10 min)
4. **Rozšírenie:** Určenie vzdialenosti a cesta späť (25 min)
5. **Vyhodnotenie:** Preskúšanie robotov a sebahodnotiaca karta (5 min)

## ZAPOJENIE (CCA 5 MIN):

Tému uvidíme príkladom zo života, kde sa majú používať úložiská na uchovanie predmetov, dát, materiálov, aby sme ich odložili na neskoršie použitie.

Námety na začatie diskusie:

1. V zoologickej záhrade potrebujú presťahovať levy na miesto tigrov, pričom vieme, že tieto šelmy sa nemajú radi, nemôžu byť v spoločnom priestore.  
Ako by ste postupovali, aby ste ich bezpečne presťahovali?
2. Jeden pohár je plný vody a v druhom rovnakej veľkosti je čaj. Ako by ste vedeli vymeniť obsah dvoch pohárov. (Môžete to demonštrovať aj reálne.)
3. Záhradníci začiatkom leta pozbierali cibule farebných tulipánov, aby na jeseň ich mohli znovu zasadiť tak, aby vytvorili záhony rôznych farieb. Skladovať ich budú v pivnici. Navrhnite ako majú postupovať pri balení, aby cibule rôznych farieb sa nepomiešali.

Žiaci majú pochopiť, že pri riešení týchto problémov potrebujú niečo na dočasné uloženie objektov. Tieto úložiská majú byť vhodné na uchovanie týchto objektov, a majú byť jednoznačne označené, aby sa uložené objekty nestratili, a aby ich pomenovanie naznačovalo ich obsah.

Učiteľ má usmerňovať diskusiu tak, aby tieto pravidlá odzneli.

## SKÚMANIE (CCA 35 MIN):

### Počítanie čiar a prekážka (Pracovný list – 1. a 2. úloha)

Učiteľ má vytvoriť označenia na podložke, alebo na podlahe triedy pomocou jednofarebnej izolačnej pásky podobne, ako je prechod pre chodcov na ceste z priečných čiar. Dĺžka čiar nech je aspoň 10 cm, a nech sú rovnobežne uložené vo vzdialenosti aspoň 5 cm, vzdialenosť medzi jednotlivými čiarami môže byť rôzna.

Žiaci majú pracovať s robotom, ktorého postavili na predchádzajúcej hodine na sledovanie dráhy.

Pomocou vzorového programu vyhotovia vlastný program, kde si nastaví parametre blokov na základe vlastného robotického modelu. Počas programovania majú odskúšať správne fungovanie robota. Nechajme ich samostatne pracovať, prípadné miskoncepce budeme riešiť v nasledujúcej fáze vysvetľovania.

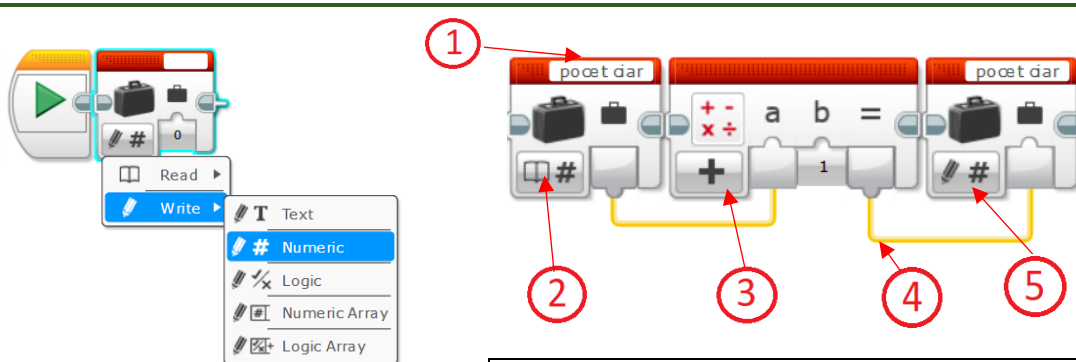
### Riešenie 1. úlohy

#### Úloha 1

Robot počíta čiary

*Naprogramujte robota, ktorý dokáže spočítať vyznačené čiary na podložke, pri zistení každej čiar aj zapísať a aktuálny počet spočítaných čiar vypíše na displej.*

*Na zapamätanie počtu čiar budete potrebovať úložisko. Medzi **červenými operačnými blokmi** nájdete kufríky (Premenné - Variables), do ktorých môžete vkladať údaje a následne aj čítať z nich.*



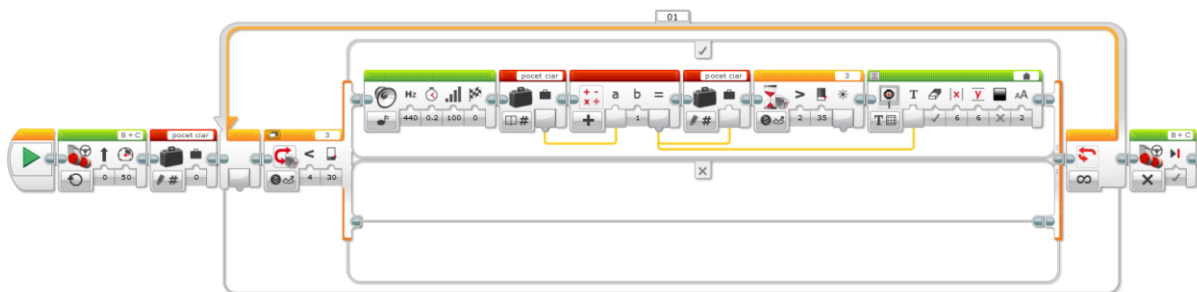
Doplňte zodpovedajúce čísla k popisom:

1	Meno premennej
2	Čítanie z premennej
5	Zápis do premennej
3	Pripočítanie čísla k hodnote uloženej v premennej
4	Dátový tok, prepojenie medzi blokmi

### Možné riešenie 2. úlohy

#### Úloha 2

Vytvorte nasledovný program s názvom **pocitanie\_ciar** a nastavte hodnoty a porty blokov tak, aby váš robot riadený týmto programom splnil zadanú úlohu.



Čakací blok s podmienkou zisťovania medzery zabezpečí, že jednu čiaru nebude počítať viackrát.

V predloženom programe v pracovnom liste počet opakovaní v cykle je nastavený na nekonečnú, robot počíta čiary, pokiaľ ho nevypneme tlačidlom na vypínanie. Autonómne zastavenie žiaci budú riešiť v nasledujúcej úlohe

### 3. úloha

Táto úloha je vylepšenie predchádzajúcej úlohy, treba zabezpečiť, aby pri určitej prekážke robot dokončil počítanie a zastavil sa.

**Prekážka** môže byť stena miestnosti, krabica, alebo čo máte k dispozícii.

Žiaci sa majú rozhodnúť, **aký senzor** budú používať. Pravdepodobne vyberú tlakový alebo ultrazvukový senzor. Nechajme ich experimentovať. Vybraný senzor majú pripevniť na prednú časť robota.

Podľa použitého senzora treba nastaviť koncovú podmienku v cykle programu, a taktiež vhodné hodnoty riadenia senzorov. Po dosiahnutí prekážky počítanie sa zastaví a vypneme motory, nastavíme blok motorov na OFF.

### Možné riešenie 3. úlohy

#### Úloha 3

Prekážka

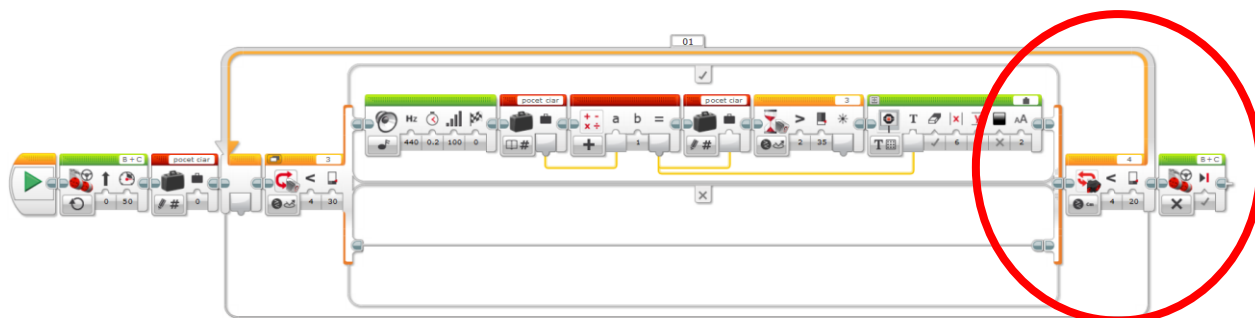
Vylepšite robota tak, že ak dôjde k prekážke, nech sa zastaví a zapíska toľkokrát, cez koľko čiar prešiel.

Na sledovanie koncovej prekážky **navrhnete použitie senzora, namontujete na robota a nastavíte riadenie cyklu** pomocou tohto senzora. Možný návod na montáž nájdete v časti Building Instructions.

Svoje riešenie zaznamenajte:

Rozhodli sme sa používať senzor:	US senzor / tlakový senzor
Dôvod nášho rozhodnutia:	Prekážku bezpečne zistí pomocou senzora

Doprogramujte opakované zapískanie po zastavení.

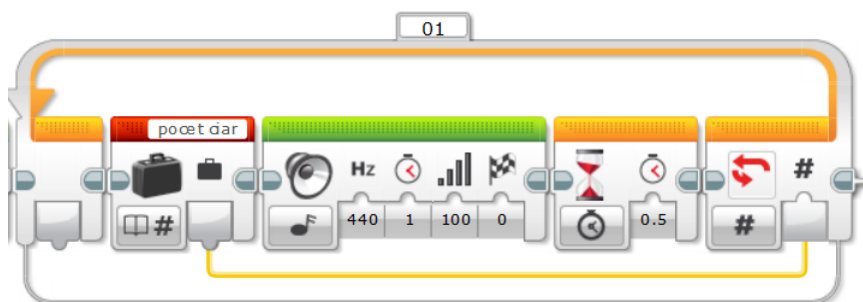


Ostáva ešte doprogramovať opakované zapískanie po zastavení.

Žiaci už na základe predchádzajúcich zistení by mali vedieť to urobiť bez problémov.

Opäť sa využíva hodnota uložená v premennej **pocet ciar**, počet opakovaní sa nastaví priamo z premennej pomocou dátového toku.

#### Možné riešenie:



Vzorové riešenie nájdete v prílohe tejto metodiky v projekte **premenne.ev3** v programe **Pocitane\_ciar** aj s komentármi.

## VYSVETĽOVANIE (CCA 10 MIN):

Najprv je potrebné slovne kontrolovať, či žiaci v prvej úlohe doplnili v tabuľke k popisom správne čísla. Postačí, ak niekto prečíta a ostatní na to reagujú. Prípadné nedorozumenia si môžu vysvetliť.

### Vysvetľovanie úlohy 1. a 2.:

Ako fungujú roboty? Čo by mohlo fungovať inak?

Žiaci majú ukázať fungovanie robota, slovne komentovať svoje riešenia a vzájomne si vysvetliť vzniknuté problémy.

Uvádzame niekoľko najčastejších chýb:

- **Blok premennej** má byť pomenovaný. Kliknutím na prázdne okienko bloku v pravom hornom rohu sa objaví ponuka Add Variable (pridaj premennú), kde treba zapísať názov premennej, môže to byť aj viac slov.
- Pri **opakovanom použití bloku premennej** v inej časti programu treba nastaviť názov premennej, kliknutím na okienko sa objaví ponuka všetkých premenných, ktoré sú definované v danom projekte.
- Pri použití bloku premennej treba nastaviť, či blok sa používa na **čítanie z premennej**, alebo na **zápis do premennej**. Taktiež ak pracujeme s číslami, má byť nastavené Numeric – numerické dáta.
- Dátové toky majú byť správne a logicky pospájané.
- **Nastavenie hodnoty farebného senzora** pri podmienke na zisťovanie čiary má byť určené na základe merania, ak sa používa meranie podľa intenzity odrazeného svetla (Color Sensor – Compare – Reflected Light Intensity) pomocou Port View zistíte aktuálne hodnoty.  
Ak sa používa rozpoznanie podľa farieb (Color Sensor – Compare - Color) treba otestovať, či v daných podmienkach senzor skutočne vidí danú farbu.
- **Výpis na displej** sa robí pomocou zeleného Display bloku, ak chceme zobraziť údaj z premennej, v pravom hornom rohu má byť nastavené **Wired** (dátový tok).
- Výpis na display sa môže upraviť pomocou nastavenia súradnicovej sústavy v bloku.
- Ak robot prechádza cez čiernu čiaru dlhšie, napríklad ak je širšia, môže sa stať, že zaznamená čiaru viackrát. Preto je vhodné **vložiť po zapísaní do premennej čakací blok**, ktorý zabezpečí čakanie na medzeru medzi čiarami.

### Vysvetľovanie úlohy 3.:

Ako zisťuje robot prekážku?

Žiaci majú predstaviť a ukázať svoje riešenia na fungujúcom modeli.

Ak senzor nemeria spoľahlivo a robot sa nezastaví pred prekážkou, odporúčame skontrolovať nasledovné:

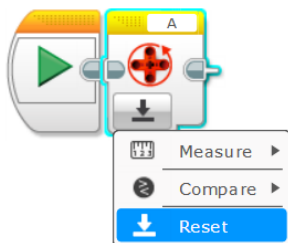
- Pri používaní ultrazvukového senzora je dôležité, aby pred senzorom sa nenachádzalo nič iné, žiadna vyčnievajúca súčiastka, čo by mohlo byť v zornom uhle senzora.
- Pri používaní dotykového senzora presnejšie zatlačenie môžeme zabezpečiť, ak vložíme do senzora tyčku, alebo iné predĺženie, ktorá skôr dosiahne prekážku.
- V programe **riadenie cyklu je ukončené podmienkou**.  
Pri ultrazvukovom senzore má byť minimálne 6 cm, lebo pri kratšej vzdialenosti senzor už nemeria spoľahlivo.
- **Zapískanie** – zvuk sa môže nastaviť pomocou výberu jednoduchých tónov, alebo je možnosť výberu aj rôznych zvukových efektov z ponuky Play File, a v pravom hornom rohu bloku sa nastaví zvuk z ponuky LEGO Sound Files. Medzi jednotlivými zvukmi je potrebné použiť čakací blok (Wait), aby sa dali rozlíšiť.

## ROZŠÍRENIE (CCA 25 MIN):

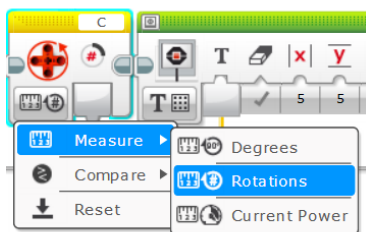
V tejto úlohe použijeme **rotačný senzor**, ktorý netreba zvlášť primontovať, lebo je to súčasťou použitého servomotora na pohon kolies.

Blok rotačného senzora sa nachádza medzi žltými senzorovými blokmi, a v programe ho nastavíme podľa motora v pravom hornom okienku.

Pri použití rotačného senzora je vhodné pred meraním senzor vynulovať (resetovať):



Otáčky je možné merať v uhloch, alebo rotáciách, nastavení v bloku senzora:



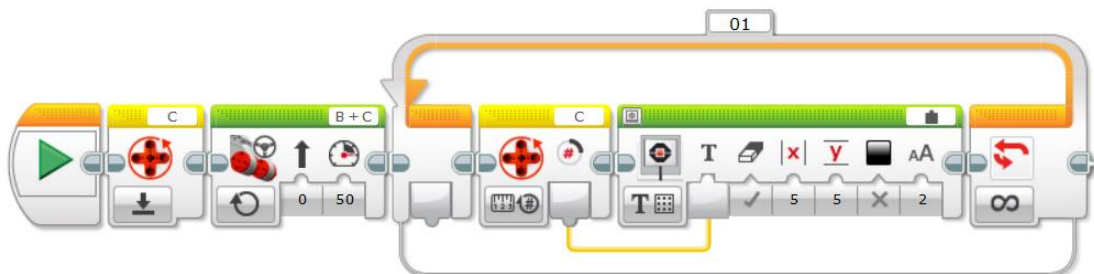
V prvej časti úlohy žiaci majú len nastaviť porty motorov a rotačného senzora v programe, a bádateľsky zistiť, ako robot zisťuje prejdenu vzdialenosť.

### Možné riešenie 4. úlohy:

#### Úloha 4 Určenie vzdialenosti

*Pomocou rotačného senzora robot dokáže sledovať akú vzdialenosť prešiel. Tento senzor je zabudovaný do servomotora, na ktorý je pripevnené koleso.*

*Vytvorte nasledovný program s názvom **vzdialenost**, kde použijete blok rotačného senzora zo žltých senzorových blokov.*



*Otestujte program pomocou robota.*

*Na základe svojich zistení navrhnete program s názvom **spiatocka**, pomocou ktorého sa robot rozbehne, zastaví sa pred prekážkou a potom sa vráti presne na štartovacie miesto.*

*Využite programové bloky rotačného senzora a blok premennej.*



Pomocou tohto zistenia žiaci by mali samostatne riešiť úlohu s názvom **spiatocka**, kde robot sa má rozbehnúť podobne ako v predchádzajúcej úlohe, pred prekážkou sa zastaviť a vrátiť sa presne na štartovacie miesto.

Je to komplexná úloha, kde využijú algoritmy z predchádzajúcich bádání.

Možné riešenie úlohy:



Keďže niektoré programové časti sa použijú z predchádzajúcich úloh, žiaci môžu použiť aj funkciu kopírovania, funguje to podobne, ako v bežných programoch, po vyznačení s CTRL+C, CTRL+V.

Všetky programy vytvorené na tejto hodine sa nachádzajú v jednom projekte, preto prepínanie medzi programami je jednoduché.

#### Poznámka pre učiteľa:

Žiaci riešili dve rôzne úlohy: počítanie čiar a zisťovanie prejdenej dráhy s možnosťou vrátenia sa na štart.

Možno žiakov to napadne skôr, ale môžete sa opýtať, či by bolo možné tieto dve úlohy spojiť do jedného riešenia.

Možné riešenie k tomu nájdete v priloženom projekte **prenenne.ev3**, program s názvom **ObeNaraz**.

### VYHODNOTENIE (CCA 5 MIN):

Žiaci predvedú svoje riešenia, predstavia fungovanie robotov, vzájomne si ich slovne hodnotia.

Na konci pracovného listu je krátka sebahodnotiaci tabuľka, ktorú vyplnia po skončení prezentácií.

Počas hodiny žiaci riešili tri rôzne úlohy:

1. Počítanie čiar
2. Zistenie prekážky
3. Vrátenie sa na štartovacie miesto

V sebahodnotiacej karte by mali viesť ohodnotiť, do akej miery sa im podarilo vyriešiť tieto úlohy.

#### Sebahodnotiaci rubrika

Otázky na hodnotenie Vašej práce:	áno	čiastočne	nie
Náš robot dokáže počítať čiaru?			
Náš robot jednoznačne zisťuje prekážku?			
Náš robot sa dokáže vrátiť na štartovacie miesto po zistení prekážky ?			