EXPEDICE VEX–9:

Průvodce pro učitele

Michal Vašíček

1. ÚVOD DO EXPEDICE

Organizace výuky:

- Výuku lze realizovat jako jeden ucelený blok, nebo ji rozdělit (tzv. spirálové učení): 6. ročník (pohyb a senzory), 7. ročník (podmínky a magnet), 8.–9. ročník (vlastní bloky a proměnné).

Časová dotace a fázování

- Z praxe doporučujeme vyčlenit si na jednotlivé fáze výuky následující časový prostor:
  - Mise 1 a Mise 2: 2 vyučovací hodiny (2× 45 minut) na každou misi.
  - Mise 3: 3 vyučovací hodiny (3× 45 minut).
  - Nutná rezerva: Počítejte vždy s 20–25 minutami navíc na finální demontáž a roztřídění součástek do boxů.

Příběhový rámec:

- Žáci se vtělují do role "Inženýra Lea", který přistál na planetě Auroria. Jejich prvním úkolem je sestavit průzkumný modul VEX GO (Code Base) a začít objevovat planetu. Po prvotním prozkoumání okolí a základním roztřídění nalezených surovin se expedice s planetou rozloučí, zanechá vzkaz v prachu a vrací se zpět na domovskou planetu Zemi.

- **Organizace rolí ve třídě:**
  - **Žák (Inženýr Leo):** Získává instrukce k postupu z pracovních listů, které fungují jako komunikace ze "Základny na Zemi".
  - **Učitel (Velitel mise):** Přebírá roli velitele, který dohlíží na průběh mise. Jeho role je spíše kontrolní a podpůrná – pomáhá inženýrům řešit technické a krizové situace a schvaluje splněné kontrolní body. Tyto momenty, kdy má žák zavolat učitele, jsou v pracovních listech jasně označeny symbolem

Jak pracovat s pracovními listy:

- Koncept výuky je postaven na třech pilířích:

1. Žák se seznamuje s novým programovacím konceptem (cyklus, podmínka, proměnná) přímo v kontextu děje.
2. Dílčí krátké úlohy, které slouží k prvotnímu kontaktu nových programátorských konceptů.
3. Komplexní úkol (MISE), kde žáci musí nabyté dovednosti spojit dohromady k dosažení cíle (např. roztřídit suroviny).

Podpora a vzorová řešení pro učitele

- Zdrojové kódy k úlohám: Odkazy na stažitelné soubory pro přímý import do VEXcode GO.
- Videoukázky řešení: Záznamy reálného pohybu robota při správně vyřešeném úkolu.

2. PLATFORMA VEX GO

Pracovní listy jsou navrženy pro práci s platformou VEX GO. Programování průzkumného modulu probíhá prostřednictvím prostředí VEXcode GO. Je možné používat webové rozhraní nebo aplikace dostupné pro různé platformy.

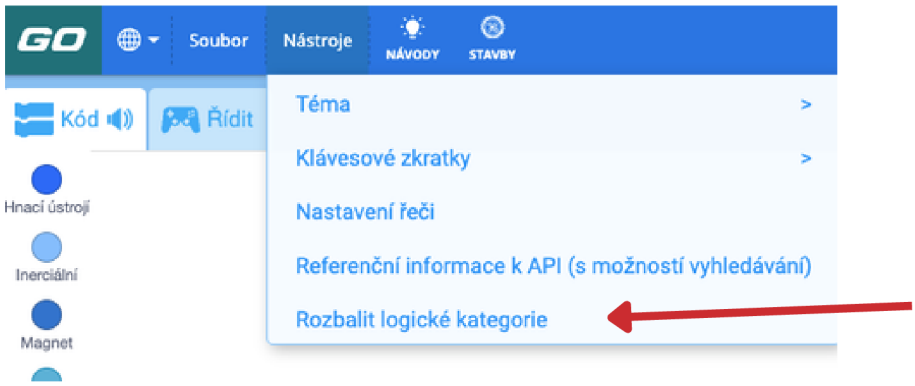
Mozek robota VEX GO se k ovládacímu panelu připojuje bezdrátově prostřednictvím Bluetooth. Je proto nezbytné, aby žáci pracovali na zařízeních (tabletech či PC), která toto připojení podporují.

UŽITEČNÉ ODKAZY:

- VEXcode GO (Webové rozhraní): codego.vex.com
- Stažení aplikací (Tablety/PC): VEXcode GO Install
- Podpora pro VEXcode GO: Get Started with VEX GO
- Návody na stavby: VEX GO Builds

TIP PRO LEPŠÍ ORIENTACI V PROSTŘEDÍ:

- V základním nastavení prostředí VEXcode GO jsou všechny kategorie logických bloků skryty pod jednou společnou záložkou "Logika" v levém bočním panelu. Některým žákům může výrazně pomoci s orientací, pokud jim tyto kategorie rozdělíte do samostatných barevných záložek. To provedete jednoduše tak, že v horní liště kliknete na Nástroje -> Rozbalit logické kategorie.



NEJČASTĚJŠÍ PROBLÉMY:

- **Baterie a podivné chování:** Je potřeba hlídat úroveň nabití baterie (pokud kontrolka svítí oranžově nebo červeně, signalizuje to nízké napětí). Vybitá baterie způsobuje nesprávné chování motorů i optického senzoru.
- **Nefungující součástky (nedocvaknutý kabel):** Často se stává, že konektor senzoru nebo motoru není úplně "docvaknutý" do portu v mozku. Pokud kabel není zasunutý nadoraz, daná součástka nebude fungovat.
- **Špatně zapojené porty:** Součástky musí být připojeny do portů přesně podle návodu. Správné zapojení je také snadno dostupné k nahlédnutí přímo v prostředí VEXcode GO, vždy podle aktuálně postaveného modelu.

3. NEŽ ZAČNEME

Následující přehled upozorňuje na kritická místa zjištěná z naší praxe, abyste předem věděli, na jaké bariéry si dát při výuce pozor a dokázali jim efektivně předcházet.

Terminologický zmatek a nekonzistence pojmů

- **Bariéra:** Střídání názvů (např. „tlačítko“ vs. „nárazník“ či „bumper“) žáky mate. U méně motivovaných jedinců to vede k rychlé rezignaci a vyhýbavému chování.
- **Prevence:** Před začátkem práce proveďte transparentní úvodní vyjasnění pojmů. Sjednoťte si s žáky terminologickou úmluvu a důsledně ji dodržujte. V pracovních listech je primárně používáno označení tlačítko.

Asymetrická kooperace ve skupině

- **Bariéra:** Jeden žák převezme roli programátora/konstruktéra, zatímco druhý je odsunut do pasivní role „vyhledávače součástek“, což snižuje jeho motivaci i efektivitu učení.
- **Prevence:** Nastavte jasná pravidla spolupráce. Zaveďte systém pravidelného střídání rolí (např. povinná výměna pozic „programátor“ a „kontrolor“ po několika krocích).

Dezorientace v hardwaru a organizační chaos

- **Bariéra:** Hromadné připojování řídicích jednotek přes Bluetooth bez označení vede k omylům (žáci se připojí k cizímu robotovi). Tím ztrácejí kognitivní kapacitu ještě před samotným programováním.
- **Prevence:** Fyzicky označte stavebnice i softwarový název Bluetooth (tzv. mozek). Přidělte konkrétní sady fixním pracovním dvojicím a přeneste na ně osobní odpovědnost za stav a roztřídění dílků.

Ztráta motivace u kognitivně rychlejších žáků

- **Bariéra:** Rychlejší žáci dokončí základní úkol dříve a při absenci podnětů nastupují prostoje a nuda. Striktní krokovací checklisty navíc omezují jejich kreativitu.
- **Prevence:** Uplatněte diferenciaci. Nadaným žákům odeberte detailní návody a zadejte jim pouze abstraktní cíl. Z metodické příručky jim operativně předkládejte rozvojové úlohy, ale až po úspěšném dokončení základní mise.

Strach z chyby a frustrace z nefunkčního kódu

- **Bariéra:** Žáci vnímají chybu v algoritmu jako osobní selhání, což je blokuje v dalším postupu.
- **Prevence:** Využívejte formativní hodnocení. Ukažte ladění chyb jako přirozenou, obvyklou a konstruktivní součást programování. Oceňujte vytrvalost, inženýrskou houževnatost a vzájemnou pomoc, nikoliv jen hotový výsledek.

4. POHYB A SENZORY

Vstupní předpoklady:

- Žáci umí zapnout tablet či počítač a orientují se v běžném webovém prohlížeči. Není vyžadována žádná předchozí zkušenost s robotikou ani programováním.

Potřeby pro výuku:

- Ohraničená plocha pro jízdu robotů (například VEX hřiště nebo robotický stůl).
- Ovládací zařízení (počítač, notebook či tablet) pro každou pracovní dvojici.
- Stavebnice VEX GO pro každou pracovní dvojici.
- Vytištěné žákovské pracovní listy (str. 2–9).

Tato lekce obsahuje:

- **Sestavení** základního hardwarového **podvozku** robota (Code Base) a připojení senzorů.
- Bezdrátové **spárování** Mozku robota s prostředím VEXcode GO přes Bluetooth.
- **Programování** základního **sekvenčního pohybu** (jízda vpřed/vzad, otáčení o přesné stupně).
- **Vyhodnocení dat ze senzorů:**
  - detekce překážek pomocí senzoru Oko (Eye sensor),
  - manuální spouštění pomocí tlačítka (LED Bumperu).
- Aplikaci nekonečné **smyčky** a **generátoru náhodných čísel**.

Přípravné úkoly:

- Test pohybu:
  - zdrojový kód řešení
  - videoukázka vyřešeného úkolu
- Otestování senzorů:
  - zdrojový kód řešení
  - videoukázka vyřešeného úkolu

MISE: Průzkum okolí:

- V této misi je úkolem žáků naprogramovat robota tak, aby projel celou ohraničenou plochu. Pokaždé, když pomocí senzoru detekuje překážku (stěnu), musí se jí vyhnout otočením o náhodný úhel a následně pokračovat dále v jízdě.
  - zdrojový kód řešení
  - videoukázka vyřešeného úkolu

Na co si dát pozor:

- Výběr správného zařízení: V kroku připojování robota je naprosto klíčové pohlídat, aby žáci v záložce zařízení opravdu vybrali a přidali model „CODE BASE“. Pokud to neudělají, nezobrazí se jim v levém menu modré bloky pro hnací ústrojí (pohyb) a nebudou moci plnit úkoly.

5. LOGIKA – ROZHODOVÁNÍ

Vstupní předpoklady:

- Žáci umí zapnout zařízení, bezdrátově spárovat Mozek robota přes Bluetooth a ovládají základní algoritmizaci z předchozí lekce.

Potřeby pro výuku:

- Plocha pro jízdu robotů.
- Ovládací zařízení (počítač, notebook či tablet) pro každou pracovní dvojici.
- Stavebnice VEX GO pro každou pracovní dvojici.
- Barevné disky ze sady VEX GO (představující suroviny).
- Vytištěné žákovské pracovní listy (str. 10–13).

Tato lekce obsahuje:

- **Rozšíření konfigurace** robota o elektromagnet.
- Sestavování **podmínek** a využití **logického větvení** programu.
- Práci s logickými výrazy na základě **rozpoznávání barev** pomocí senzoru Oko.
- Manipulaci s externími objekty (zvedání a pokládání barevných disků).

Přípravné úkoly:

- Test elektromagnetu (manipulace s objektem):
  - zdrojový kód řešení
- Otestování senzoru (rozpoznání barvy):
  - zdrojový kód řešení

MISE: Třídění surovin:

Robot musí dojet k surovině (barevnému disku), zvednout ji pomocí elektromagnetu, pomocí senzoru Oko identifikovat její barvu a na základě logické podmínky (pokud je barva X, udělej toto) surovinu správně roztřídit a odvézt na příslušné místo.
Před spuštěním mise stanovte přesný počet surovin, které bude potřeba vytřídit, a připravte je naskládané za sebou do jedné linie.

- zdrojový kód řešení
- videoukázka vyřešeného úkolu

Na co si dát pozor:

- Logické větvení programu a sestavování podmínek je abstraktní koncept, u kterého žáci často selhávají, pokud se jim nedostane vysvětlení předem. Dohlédněte, aby žáci nepřeskočili teoretickou pasáž v pracovním listu, která fungování podmínek vizuálně a procedurálně vysvětluje.

6. PROMĚNNÉ A VLASTNÍ BLOKY

Vstupní předpoklady:

- Žáci umí spárovat zařízení přes Bluetooth a ovládají sekvenční programování i logické větvení z předchozích lekcí.

Potřeby pro výuku:

- Plocha pro jízdu robotů.
- Ovládací zařízení (počítač, notebook či tablet) pro každou pracovní dvojici.
- Stavebnice VEX GO pro každou pracovní dvojici.
- Vytištěné žákovské pracovní listy (str. 14–18).

Tato lekce obsahuje:

- Práci s abstraktnějšími koncepty jako je časovač a ukládání dat do proměnných.
- Vytváření vlastních bloků (uživatelských funkcí).
- Aplikaci hierarchické abstrakce – vnořování funkcí do nadřazených celků.

Úkoly:

- Vytvoření vlastního bloku (manévr mávnutí):
  - zdrojový kód řešení
  - videoukázka vyřešeného úkolu
- Využití proměnné a časovače:
  - zdrojový kód řešení
  - videoukázka vyřešeného úkolu

MISE: Rozloučení (Kódování znaků):

- V této misi žáci simulují robota jako automatické písátko. Úkolem je naprogramovat řídicí modul tak, aby správně převáděl manuální počet stisků tlačítka na konkrétní vykreslení příslušného písmene do reálného pohybu robota na podložce. K dosažení tohoto cíle si žáci tvoří vlastní bloky (uživatelské funkce) pro vykreslení jednotlivých, izolovaných písmen.
  - zdrojový kód řešení
  - videoukázka vyřešeného úkolu (verze s mezerami)

Na co si dát pozor:

Koncepty jako časovač nebo proměnné jsou silně abstraktní. Žáci zde často selhávají, pokud přeskočí teorii. Dohlédněte, aby nevynechali teoretické pasáže v pracovním listu, které tyto bloky krok za krokem vysvětlují. Případně nabídněte další podporu.

7. NAVAZUJÍCÍ ÚKOLY PRO MISI 1

Optimalizace průzkumu
Základní pohyb už náš průzkumný modul zvládá, ale je to ta nejrychlejší cesta?

Tvým úkolem je najít nejefektivnější a **nejrychlejší strategii**, jak kompletně prozkoumat celý sektor.

Vezmi si stopky (nebo mobil) a měř čas. Testuj a porovnávej různé přístupy v kódu:

- zkus vyměnit otáčení o náhodný úhel za fixně stanovený úhel,
- nebo naprogramuj systematické přejíždění plochy pruh po pruhu.

Změř a vyhodnoť, který postup je pro náš prostor absolutně nejlepší.

Optimalizace průzkumu
Základní pohyb už náš průzkumný modul zvládá, ale je to ta nejrychlejší cesta?

Tvým úkolem je najít nejefektivnější a **nejrychlejší strategii**, jak kompletně prozkoumat celý sektor.

Vezmi si stopky (nebo mobil) a měř čas. Testuj a porovnávej různé přístupy v kódu:

- zkus vyměnit otáčení o náhodný úhel za fixně stanovený úhel,
- nebo naprogramuj systematické přejíždění plochy pruh po pruhu.

Změř a vyhodnoť, který postup je pro náš prostor absolutně nejlepší.

Optimalizace průzkumu
Základní pohyb už náš průzkumný modul zvládá, ale je to ta nejrychlejší cesta?

Tvým úkolem je najít nejefektivnější a **nejrychlejší strategii**, jak kompletně prozkoumat celý sektor.

Vezmi si stopky (nebo mobil) a měř čas. Testuj a porovnávej různé přístupy v kódu:

- zkus vyměnit otáčení o náhodný úhel za fixně stanovený úhel,
- nebo naprogramuj systematické přejíždění plochy pruh po pruhu.

Změř a vyhodnoť, který postup je pro náš prostor absolutně nejlepší.

8. NAVAZUJÍCÍ ÚKOLY PRO MISI 2

Detekce kontaminace a bezpečnostní limit
Základna hlásí problém! Jedna ze surovin (modrá) byla právě překlasifikována na „toxický vzorek“.

Z bezpečnostních důvodů dokáže náš třídicí modul během jedné jízdy zpracovat a přesunout **maximálně dva** tyto nebezpečné objekty.

Uprav svůj stávající algoritmus. Musíš do něj zapojit počítání (proměnnou) tak, aby po úspěšném vytřídění druhého toxického disku robot okamžitě zastavil a vyčkal na místě.

Detekce kontaminace a bezpečnostní limit
Základna hlásí problém! Jedna ze surovin (modrá) byla právě překlasifikována na „toxický vzorek“.

Z bezpečnostních důvodů dokáže náš třídicí modul během jedné jízdy zpracovat a přesunout **maximálně dva** tyto nebezpečné objekty.

Uprav svůj stávající algoritmus. Musíš do něj zapojit počítání (proměnnou) tak, aby po úspěšném vytřídění druhého toxického disku robot okamžitě zastavil a vyčkal na místě.

Detekce kontaminace a bezpečnostní limit
Základna hlásí problém! Jedna ze surovin (modrá) byla právě překlasifikována na „toxický vzorek“.

Z bezpečnostních důvodů dokáže náš třídicí modul během jedné jízdy zpracovat a přesunout **maximálně dva** tyto nebezpečné objekty.

Uprav svůj stávající algoritmus. Musíš do něj zapojit počítání (proměnnou) tak, aby po úspěšném vytřídění druhého toxického disku robot okamžitě zastavil a vyčkal na místě.

9. NAVAZUJÍCÍ ÚKOLY PRO MISI 3

Hierarchická abstrakce (Tvorba slov)
Modul už umí kreslit izolovaná písmena. Nyní ho naučíme psát rovnou celá slova (např. „AHOJ“).

- Nejprve si vytvoř nový vlastní blok pro univerzální „mezeru“ (bezpečný posun robota do strany), aby se ti znaky nepřekrývaly.
- Následně tyto menší stavební kameny (bloky pro písmena a blok pro mezeru) poskládej a vnoř do úplně nového, nadřazeného bloku, který pojmenuješ například „slovo_AHOJ“.

V hlavním programu pak celou akci spusť tak, že po stisku tlačítka zavoláš už jen tento jediný nadřazený blok.

Hierarchická abstrakce (Tvorba slov)
Modul už umí kreslit izolovaná písmena. Nyní ho naučíme psát rovnou celá slova (např. „AHOJ“).

- Nejprve si vytvoř nový vlastní blok pro univerzální „mezeru“ (bezpečný posun robota do strany), aby se ti znaky nepřekrývaly.
- Následně tyto menší stavební kameny (bloky pro písmena a blok pro mezeru) poskládej a vnoř do úplně nového, nadřazeného bloku, který pojmenuješ například „slovo_AHOJ“.

V hlavním programu pak celou akci spusť tak, že po stisku tlačítka zavoláš už jen tento jediný nadřazený blok.

Hierarchická abstrakce (Tvorba slov)
Modul už umí kreslit izolovaná písmena. Nyní ho naučíme psát rovnou celá slova (např. „AHOJ“).

- Nejprve si vytvoř nový vlastní blok pro univerzální „mezeru“ (bezpečný posun robota do strany), aby se ti znaky nepřekrývaly.
- Následně tyto menší stavební kameny (bloky pro písmena a blok pro mezeru) poskládej a vnoř do úplně nového, nadřazeného bloku, který pojmenuješ například „slovo_AHOJ“.

V hlavním programu pak celou akci spusť tak, že po stisku tlačítka zavoláš už jen tento jediný nadřazený blok.