

VEX 123[®]

Sbírka výukových materiálů pro školy

Mgr. Milan Šatra

Základní škola Aš, Kamenná 152



Anotace

Tato sbírka výukových materiálů vznikla v průběhu prvního roku realizace výuky informatiky podle revidovaného RVP během školního roku 2021/22 na Základní škole Aš, Kamenná 152, okres Cheb. Jedná se o ucelený komplet příprav na výuku povinného předmětu **Informatika** ve 4.-9. ročníku. Příklady jsou záměrně voleny tak, aby zpravidla navazovaly na učivo vzdělávací oblasti Algoritmizace a programování (Scratch pro 1. a 2. stupeň). Žáci si na jednoduché robotické učební pomůcce ukazují a ověřují algoritmické postupy z učiva Scratch. Programováním robota VEX 123 se žáci seznámí s možnostmi programování „věcí“, které obsahují podobná technická zařízení (např. robotický vysavač).

Poděkování

ředitelce školy, Mgr. Evě Horáčkové,

která umožnila nákup tří školních sad VEX 123, vznik a ověření obsahu této sbírky

mé manželce, Mgr. Zdeňce Šatrové,

za trpělivé tolerování využití spousty společného času potřebného ke vzniku této sbírky

OBSAH

Jak mě VEXík dostal	4
Metodická doporučení	9
VEX 123 se představuje	10
Programujeme pomocí tlačítek	13
Programujeme magnetickým kóděm	29
Programování blokovými příkazy	52
Užitečné odkazy	117

Dle ročníků:

4. ročník	12
5. ročník	28
6. ročník	51
7. ročník	74
8. ročník	90
9. ročník	101

Jak mě VEXík dostal 😊

Jsem učitelem informatiky. Nové informatiky. Informatiky podle revize RVP. Co s tím budu dělat? Procházím si odborné webináře pro jednotlivé oblasti obsahu nové informatiky. Po prvotním šoku, co to po mně budou chtít, se mi to ale začíná líbit. Nejvíc mne oslovila oblast algoritmizace a programování. Autoři z JČU ji připravili parádně. Chce to ale něčím zpestřit a ukázat žákům, že tyhle věci fungují i ve skutečnosti, v reálném světě mimo programovací prostředí Scratch. Jak na to, co s tím? Jasně – chce to nějakého robota! Ale jakého, nabídka je velká, prodejci se doslova vydali na hon na školy a snaží se přesvědčit, že právě jejich robot je ten nejlepší a nejúčasnější. Bohužel na naší škole narážejí na tuhý odpor – moje ředitelka se na tato témata odmítá s prodejci bavit sama beze mne a já mám tvrdou hlavu. Chci jen to, co sám považuji za nejlepší variantu. A tak se vydávám na virtuální obhlídku nabídky robotů.

Nabídka je opravdu pestrá – malé a velké, levné, drahé a předražené, jednorázové a univerzální, stavebnice a „hotovky“. Přemýšlím o tom, k čemu vlastně chci případně roboty používat a kolik na to budu mít času. První rozhodnutí je celkem rychlé a vyřazuje kompletní skupinu stavebnicových robotických pomůcek. V nové informatice je tolik učiva, že nemám čas (a vlastně ani chuť) se zdržovat stavebnicemi... Než něco postavíme, než to naprogramujeme, než to rozebereme... – a co budu dělat v té chvíli v paralelce? Docela s radostí se zbavuji LEGA, nemusím ho.



Probírám tzv. hotovky – kompletní roboty, které nemusíme stavět, jen programovat. Pro mne hlavní výhoda? Jednoznačně se zbavuji funkce „hlídače a počítače“ kostiček. Můžu se stejnou pomůckou učit dvě hodiny po sobě v obou našich paralelkách a nemusím řešit dva různé témačky. V kabinetu mám 24 micro:bitů, které jsou sice super, ale chce to k tomu ještě něco akčnějšího. BeeBot, BlueBot, Pro-Bot, mBot, Fable, iRobot Root a spoustu dalších... Co vybrat, abych za tyhle nadstandardní pomůcky nezaplátil zbytečně moc peněz? Musím myslet na problém rozvoje digitálních kompetencí a s roboty mi je třeba na dějáku nebo zemáku asi kolegyně moc rozvíjet nebudou. Takže to chce něco univerzálnějšího, co se dát využít na prvním i druhém stupni. Navíc něco rozumně velkého (malého), aby se to vešlo na desku běžné lavice, jinak bych musel se žáky buď do družiny, nebo na zem. ...Výběr se podstatně zužuje.

A pak to přišlo! Bylo to jako blesk z čistého nebe! Jako rána pěstí mezi oči! Lásku na první pohled. Konečně jsem ho našel: **VEX 123**. Ihned jsem si ho přejmenoval na VEXíka. Je to roztomilé pojmenování. Ale on sám je roztomilý – jednoduché ovládání, univerzální použití, velké možnosti využití při výuce.

VEXík je roztomilá kombinace BeeBot a iROBOT Root. Lze ho programovat pomocí tlačítek na jeho povrchu, dále pomocí speciálního programátoru magnetickými štítky a konečně také pomocí blokového jazyka VEXCode, velmi podobného Scratch. Protože jsem nechtěl kupovat „zajíce v pytli“, požádal jsem prodejce o zapůjčení nebo o osobní návštěvu. Což se také stalo a já jsem svého vyhlédnutého „miláčka“ konečně spatřil ve skutečnosti a mohl ihned začít experimentovat. První, co musel VEXík podstoupit, byl můj univerzální test přesnosti robota – 4x (DOPŘEDU -> VPRAVOVBOK). Pokud se po tomto testu robot nevrátí po opsání čtverce, je to pro mne nepoužitelný krám.

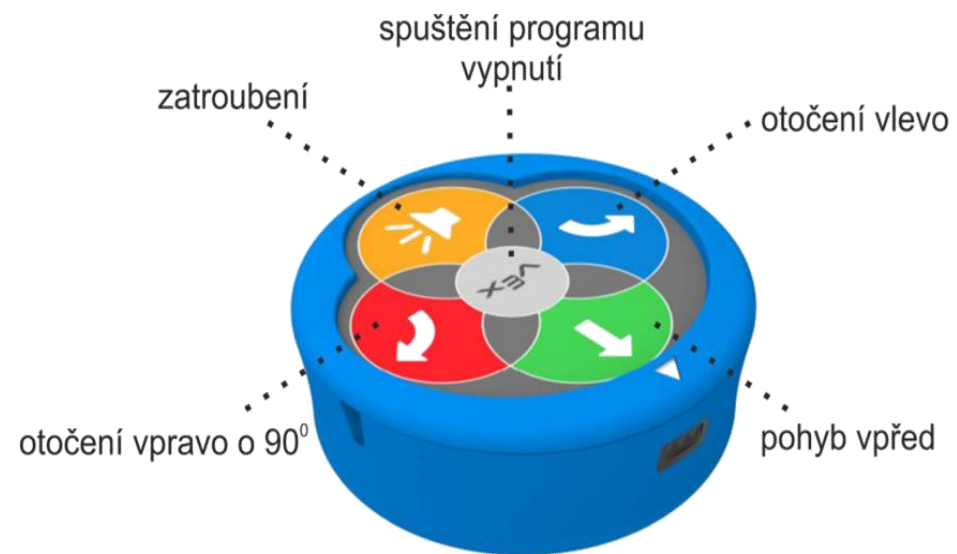
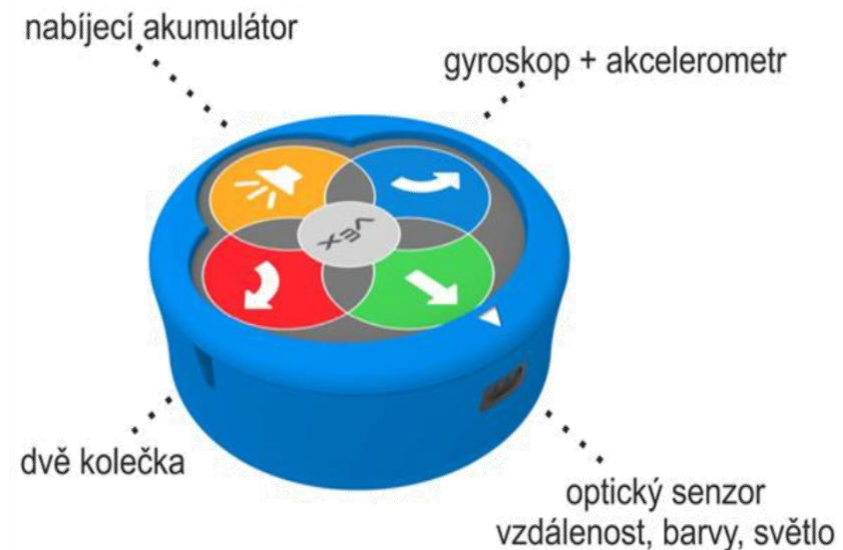
A jak dopadl VEXík? Vzhledem k tomu, že čtete tyto řádky, tak můj test zvládl VEXík na výbornou. Na výrobcem dodané herní desce se čtvercovou sítí se pohyboval mimořádně jistě a přesně. SUPER! Tak teď už jen zajít za ředitelkou, správně odůvodnit potřebu nákupu a hurá objednávat. Pořídil jsem rovnou tři školní sady VEX 123 Classroom!



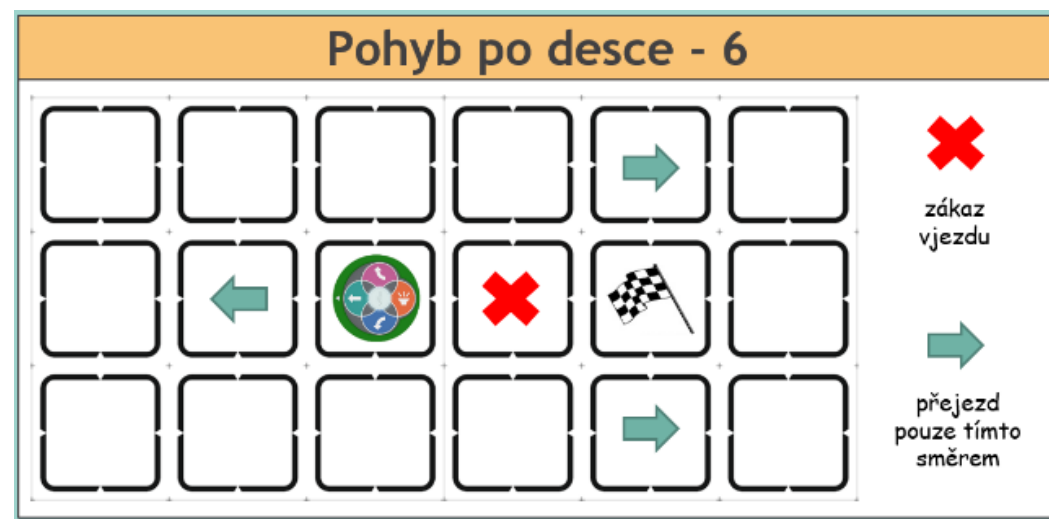
Výše uvedené sady dorazily celkem rychle a já se vrhnul na vybalení a zprovoznění dodaných VEXíků a jejich příslušenství. Zprovoznění VEXíků samotných v „tlačítkovém“ režimu bylo velmi jednoduché – stačilo zkontrolovat připojení USB-C nabíjecích kabelů a celou krabičku se 6 robůtky zapojit do zásuvky.

Bezva, VEXíci se nabíjejí, co dál? Instalace androidí aplikace VEX Class. Appka našla všechny VEXíky a já jednoho po druhém aktualizoval a přejmenoval (stejně jméno jsem jim také napsal fixem na spodek těla).

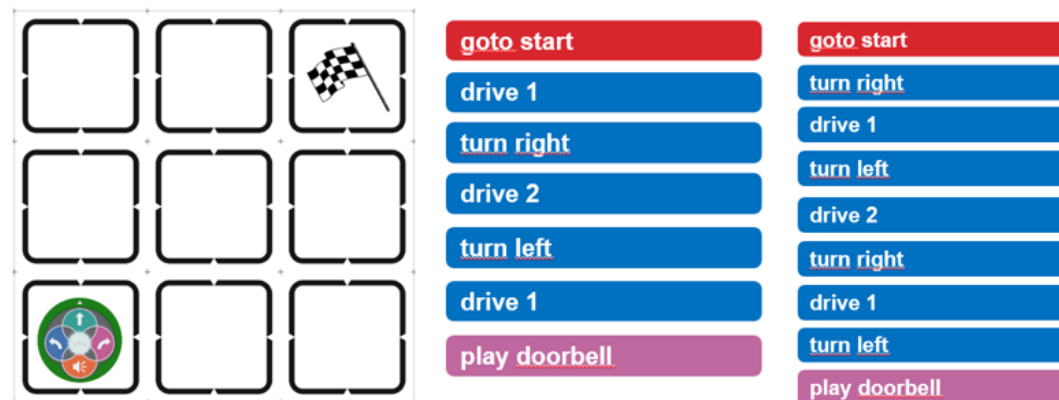
Nyní přišla správná chvíle na první experimentování. Začal jsem prohlídkou těla. Na zádech 5 tlačítek, jedno zároveň umí svítit. Vepředu optický senzor, tzv. oči. Dole další optický senzor – umí zaregistrovat čáru. Bohužel ji ale neumí sledovat jako OzoBot, takže tenhle senzor je mi zatím celkem k ničemu. Dále dvě kolečka, uvnitř krokový motůrek a gyroskop. Celkově prostě parádní jednoduchá, ale celkem „nabušená“ hračka:



Takže teď vymyslet nějaké aktivity pro žáky. S VEXíkem lze v režimu programování tlačítka vytvářet pouze sekvence. Co tedy vymyslet pro použití na jedné nebo dvou spojených herních deskách? Jasně, nějaká jednodušší „bludiště“ a pohyby napříč deskou. To není zase tak složité, chce to jen trochu fantazie. První výtvořky mám hotové, otestované a teď hurá mezi děti. Sekvence pomocí tlačítek dáme do čtyřek. Mohlo by to dát odhadem 5-6 hodin, to by mělo stačit, musí se zvládnout spoustu jiné výuky. Vše jsem vysvětlil vyučujícím informatiky ve 4. třídách a jdu se podívat na průběh hodiny. Děti jsou nadšené, bohužel se najdou i tací experti, pro které jsou sekvence skoro nad jejich síly... ☹️. Ale celkově prostě VEXík ve čtyřkách pecka.



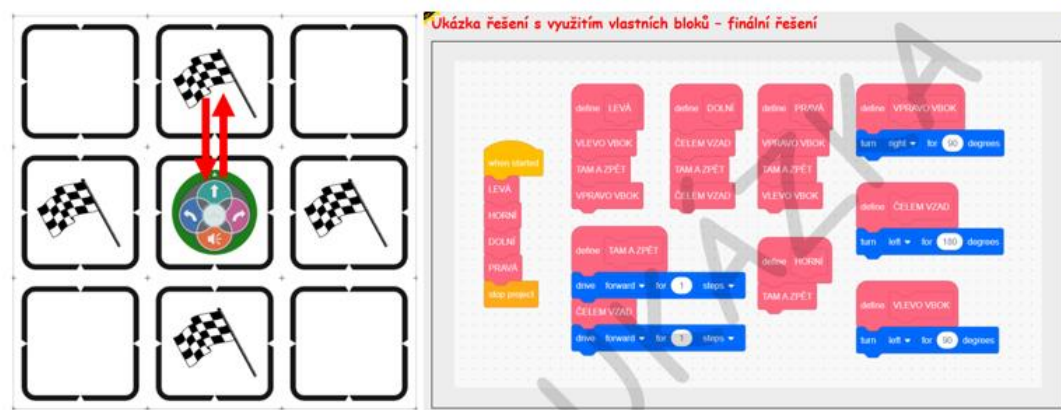
A jdeme dál! Vrhám se na černé programovací destičky – budu jim říkat programátory. Po bližším ohledání zjišťuji, že musím rozšroubovat na zádech šuplíček pro baterie a pak mne čeká opět appka VEX Class na mém mobilu, protože i programátory musím přejmenovat a aktualizovat.



Současně s tím rovnou provádím párování s VEXíky a popisují je na spodní straně tak, abychom vždy jednoduše našli pár VEXík + programátor. Práce ubíhá, za hodinu jsem se všemi 18 VEXíky hotov. A celý zvědavý se pouštím do testování programování pomocí magnetických programovacích kartiček. Opět mi to přijde jako super bezva nápad. Program se sestavuje ve vizuální podobě, žáci mohou sledovat, jak program probíhá, dokonce je k dispozici i krokování. Prostě paráda. Nadšení mi trochu kalí zjištění, že jsem narazil na první jisté VEXíkové omezení při výuce. K dispozici je pouze 10 slotů, tam to taková sláva s tvorbou programů nebude, bude to chtít předem cíleně spočítat, zda jsou úlohy řešitelné deseti kartičkami.

Takže zase stejný postup – vymyslet úlohu, otestovat, předvést vyučujícím a tentokrát hurá do pětěk. Tvorbu programu pomocí programátoru určují do 5. ročníku. Žáci jsou opět nadšení, i když máme zase některé slabší jedince. Tento problém podobně jako o ročník níže vyřeší rozdělení žáků do vhodných dvojic. Opět se dostáváme zhruba na 5-6 hodin, což by zase vzhledem k množství ostatního učiva mělo stačit.

Čeká mne poslední způsob programování VEXíka – online prostředí VEX Code, alias jiná verze Scratch. Dostupná je pouze angličtina, ale vzhledem k omezené slovní zásobě příkazů to není na závadu. Žáci 2. stupně by to měli v pohodě pobrat. Dalším bodem je volba vhodné ovládací techniky.



Mobil je malý, tablet nic moc, ideální řešení jsou naše školní chromebooky. Zapnu VEXíka, po chvíli ho chromebook detekuje, potvrdím spárování a můžu začít.

Po několika hodinách testování zjišťuji, že se mi otevírá svět úžasných možností, jak VEXíka programovat. Pro využití optických senzorů nechám kolegu dílnáře vyrobit hromadu dřevěných kvádrů o rozměrech 4x4x1 cm, které po večerech obarvuji na velkých protilehlých plochách barevnými fixy. Opět se snažím využít herního pole 1-2 desek. Je to paráda, o nápady kupodivu nemám nouzi. Dá se říct, že z jednoho úkolu drobnými obměnami vytvářím další varianty. Vytvářím si postupně prezentaci se zadáními a řešeními, je to vlastně i forma příprav. Vypadá to, že už jsem někde u 20-25 hodin výuky. Což při mé představě 6-7 hodin ročně pokrývá v pohodě celý 2. stupeň. Beru VEXíky do šestek na novou informatiku, kde se mi daří navázat na již zvládnuté dovednosti ve Scratchi. Vše jde dobře, žáci nadšení, práce opět ve vhodných dvojicích. Ještě VEXíky ukazují devátákům v rámci „staré“ informatiky – i ti jsou nadšení a mizí dokonce jejich tradiční nezájem o cokoli, co se děje ve škole po přijímačkách.

Protože jsem aktivní v metodických strukturách NPI a SYPO, začínají mne kontaktovat kolegové z celého kraje právě s dotazy, co s robotikou. Nemám na výběr, co bych měl kolegům doporučovat – VEXík mne svými možnostmi prostě fakt dostal. Ideální pomůcka na hodiny nové informatiky pro 4.-9. ročník. Začínám pořádat prezenční semináře napříč krajem a zdá se, že se náhaza VEXíkem úspěšně šíří 😊.

Metodická doporučení

Robotika by měla být vhodným praktickým doplňkem učiva oblasti algoritmizace a programování výuky informatiky ve všech ročnících. Vzhledem k tomu, že je nutné v každém ročníku do výuky zařadit všechny 4 vzdělávací oblasti informatiky, osvědčilo se mi zařazení robotické učební pomůcky VEX 123 následujícím způsobem:

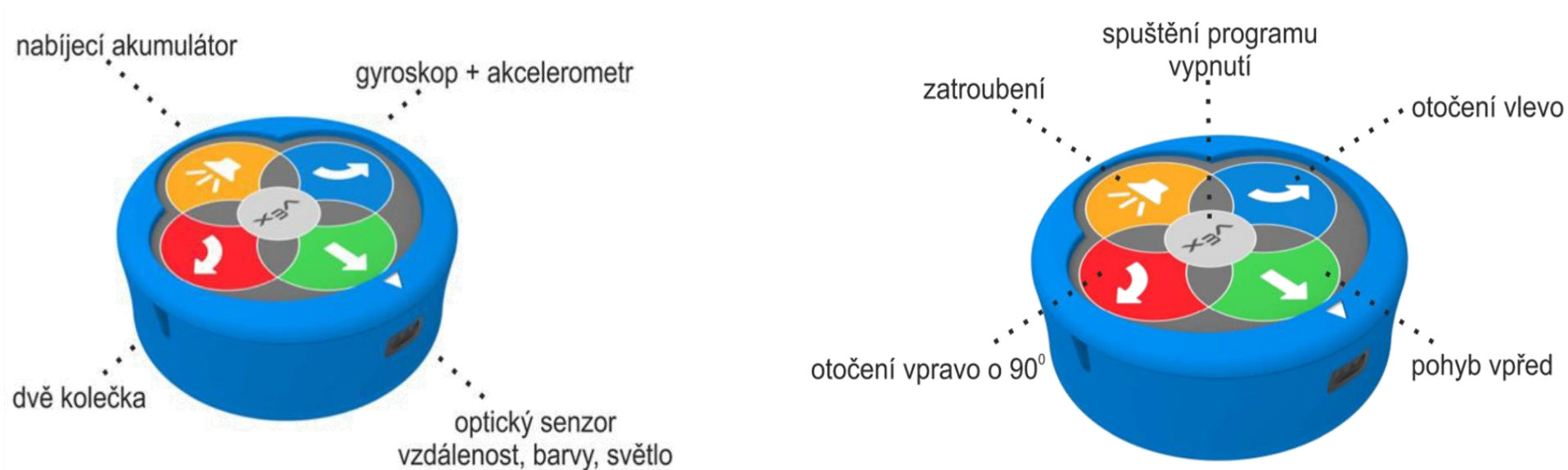
- **4. ročník** – motivační učivo před zahájením výuky algoritmizace v prostředí Scratch – 6 hodin
- **5. ročník** – po ukončení výuky algoritmizace v prostředí Scratch – 6 hodin
- **6. ročník** – po ukončení výuky v prostředí Scratch (1.-3. kapitola) – 6 hodin (sekvence + vlastní bloky)
- **7. ročník** – po ukončení výuky v prostředí Scratch (4.-6. kapitola; podmínky + cykly s podmínkou + práce s optickými senzory)
- **8. ročník** – po ukončení výuky v prostředí Scratch (7.-8. kapitola; komplexní využití, vždy s opakováním ukázek z předchozího školního roku)
- **9. ročník** – po ukončení výuky v prostředí Scratch (9.-10. kapitola; komplexní využití, vždy s opakováním ukázek z předchozího školního roku)

Přeji všem kolegyním a kolegům, které VEXík dostane stejně jako mne, hodně zábavy a úspěchů při jejich využívání. Doufám, že jim bude tato sbírka šikovným průvodcem.

Mgr. Milan Šatra, Aš, 2022

VEX 123

se představuje:



VEX 123 lze programovat třemi způsoby:

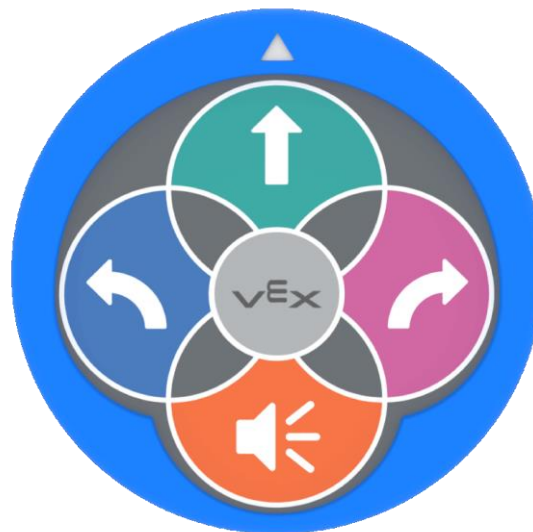
- 1) tlačítky na těle robota
- 2) magnetickým kóděrem
- 3) blokovými příkazy

Následující části se věnují vždy jednomu způsobu programování.

4. ročník

vEx 123

Programujeme pomocí tlačítek



Základní ovládání VEXíku

zapnutí
posunutí
směrem vpřed



smazání programu
zatřesení

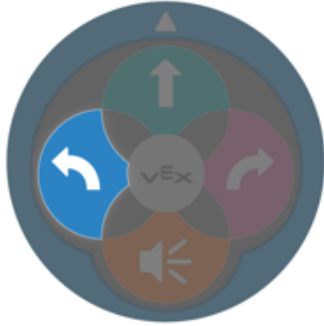
vypnutí
podržení středového
tlačítka na 3 s



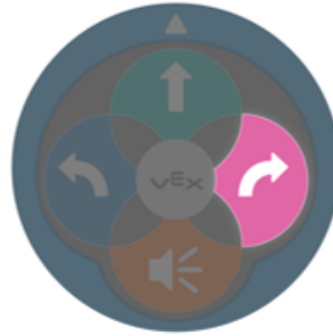
Metodická doporučení:

- zařazení jako motivační učivo před zahájením výuky Scratch (Učebnice pro 5. ročník – Modul 1)
- prvotní je důkladné seznámení žáků se základním ovládáním VEX 123

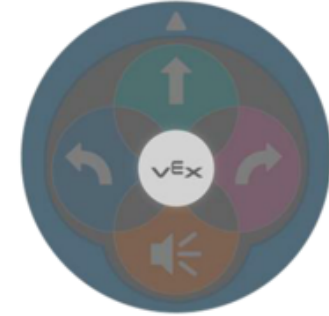
VEXíkovi základní příkazy



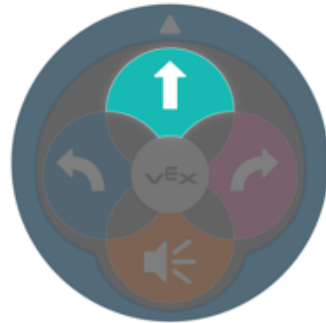
VLEVO VBOK
otočí se o 90°



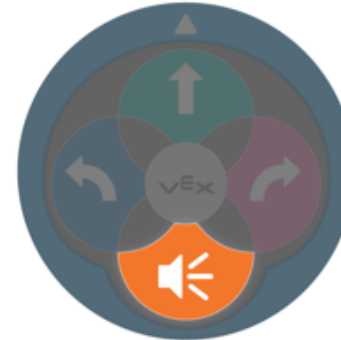
VPRAVO VBOK
otočí se o 90°



POPOJEĎ
popojede o 7,5 cm



ZAHRAJ
zahraje melodii

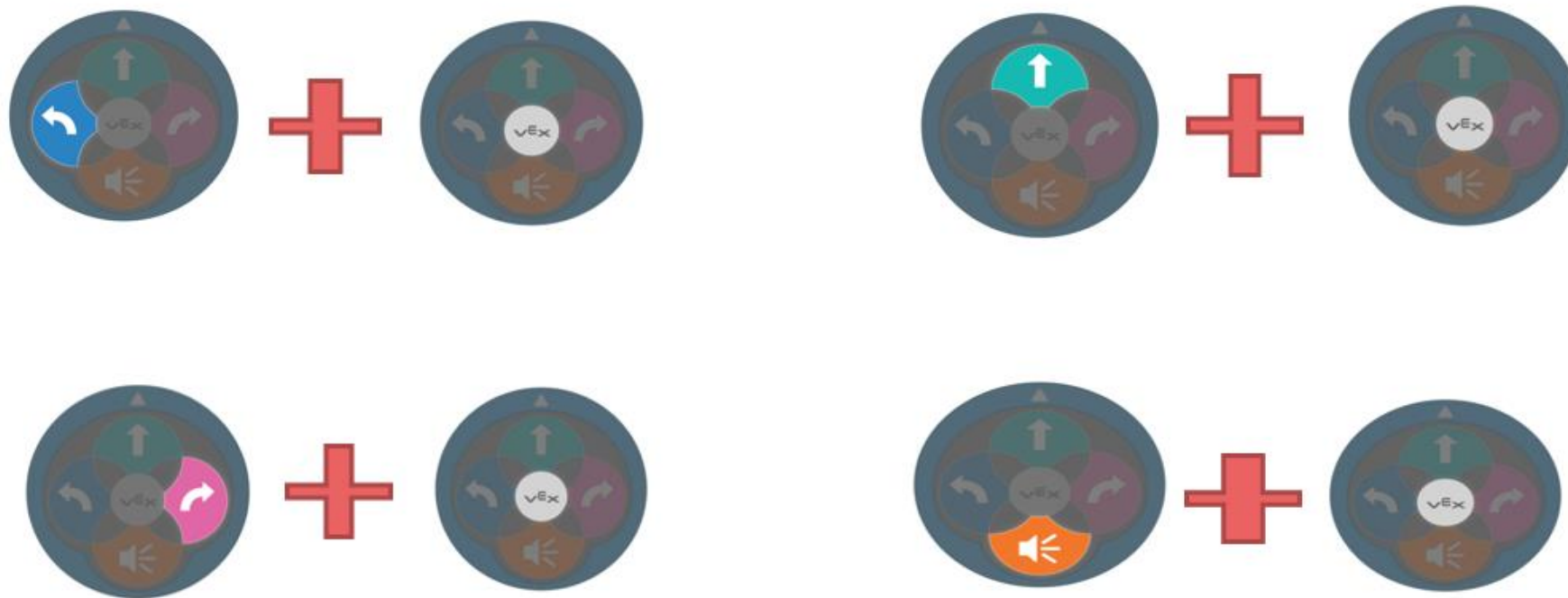


**ULOŽ /
SPUŠŤ**
zapamatuje ne
spustí program

Metodická doporučení:

- je žádoucí po žácích vyžadovat jednotnou terminologii základních příkazů

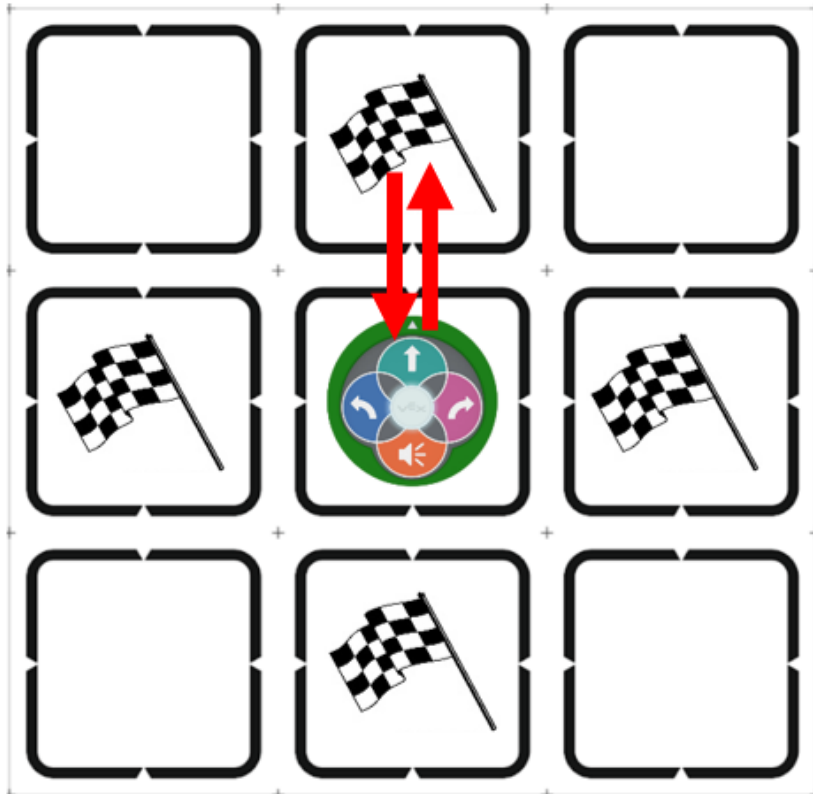
PRVNÍ AKCE



Metodická doporučení:

- pro tvorbu programu je nutné se žáky důkladně nacvičit správné naprogramování jednotlivých základních příkazů
- všechny uvedené příkazy je vhodné vyzkoušet na 1 herní desce 3x3 pole
- žáci pracují zásadně ve dvojicích – podpora spolupráce a vzájemné kontroly

Jednoduché pohyby po desce - 1



Vytvoř program, s jehož pomocí VEXík postupně dojde na cílová pole s vlajkou a zpět. Nejprve vše vyzkoušej na desce s maketou a program si zapiš do programovací tabulky. Nakonec vyzkoušej VEXíka naprogramovat a všechny úkoly splň s ním.

VEX 123 – programovací tabulka

Metodická doporučení:

- žáci mají poměrně velké problémy s představou správné posloupnosti zadávaných příkazů, proto je vhodné vyrobit tzv. programovací tabulku (v libovolném textovém editoru nebo tabulkovém procesoru) a po vytisknutí ji zalaminovat
- žáci programy zapisují stíratelnými fixy

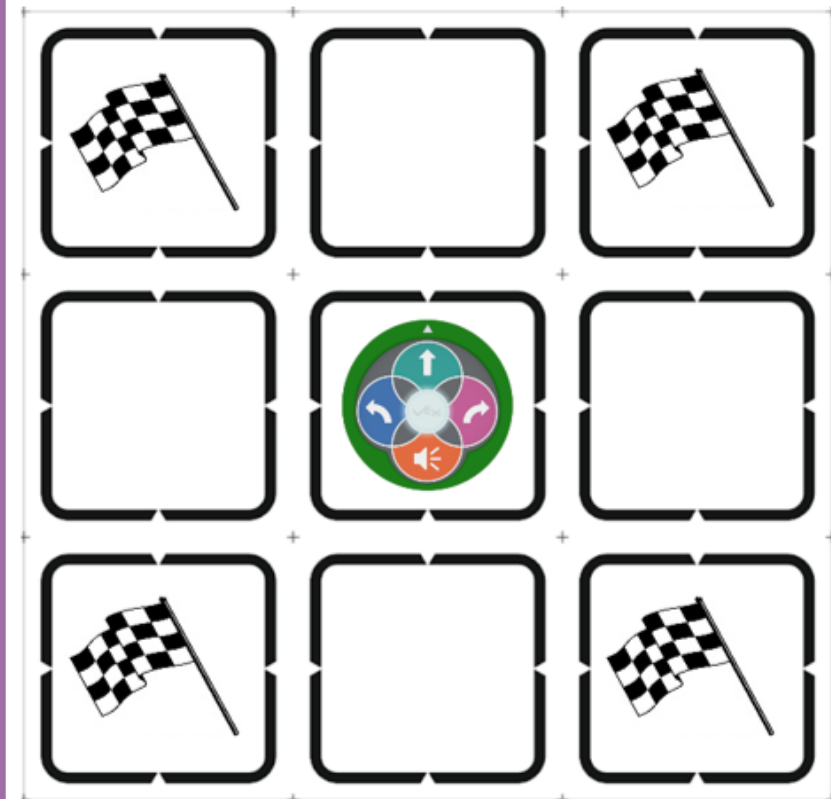
Jednoduché pohyby po desce - 1

- Čím se liší jednotlivé programy?
- V čem jsou jednotlivé programy stejné?
- Jde zápis programů nějak zjednodušit?
- Daly by se nahradit všechny programy jediným z nich?
- Do programovacích tabulek si zapiš cesty na všechny pole a pak postupně kombinuj např. nahoru + dolů, doleva + nahoru, atd. Musíš si psát pokaždé nové programy?

Metodická doporučení:

- je žádoucí žáky motivovat k systematickosti v zápisech jednotlivých programů tak, aby odhalili jejich společnou část
- VEXík se vždy po ukončení programu vrací na původní pole do původního směru
- následně lze využít pouze jeden společný program se správným pootočením na začátku a konci programu

Jednoduché pohyby po desce - 2



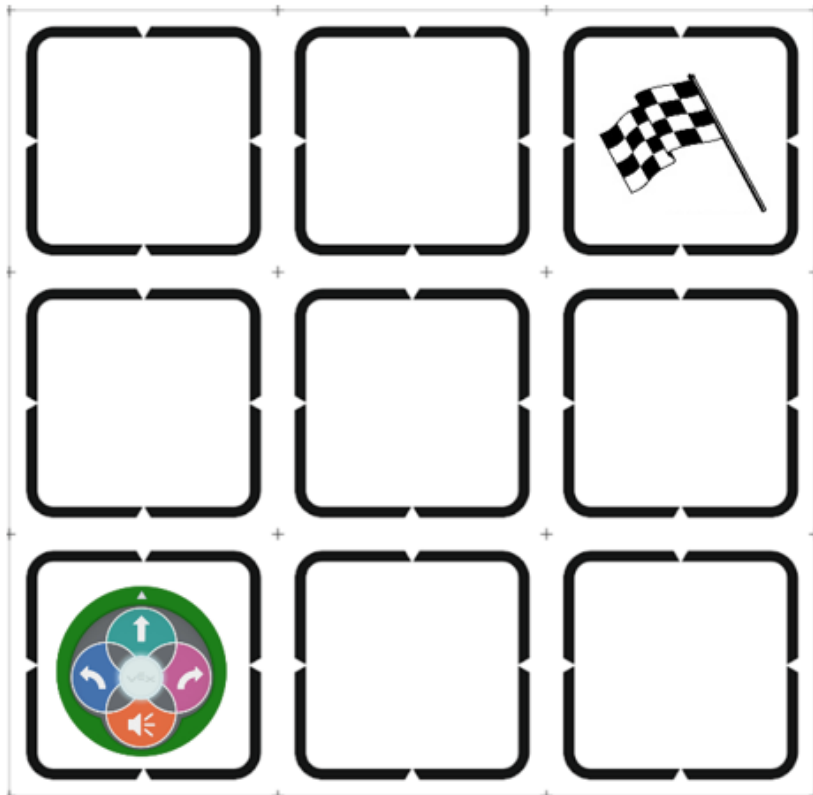
Vytvoř program, s jehož pomocí VEXík postupně dojde na cílová pole s vlajkou a zpět. Nejprve vše vyzkoušej na desce s maketou a program si zapiš do programovací tabulky. Nakonec vyzkoušej VEXíka naprogramovat a všechny úkoly splň s ním.

VEX 123 – programovací tabulka

Metodická doporučení:

- alternativa k předchozímu příkladu, jen posunuté vlajky
- pro zadání lze využít stejných otázek jako u příkladu 1

Jednoduché pohyby po desce - 3



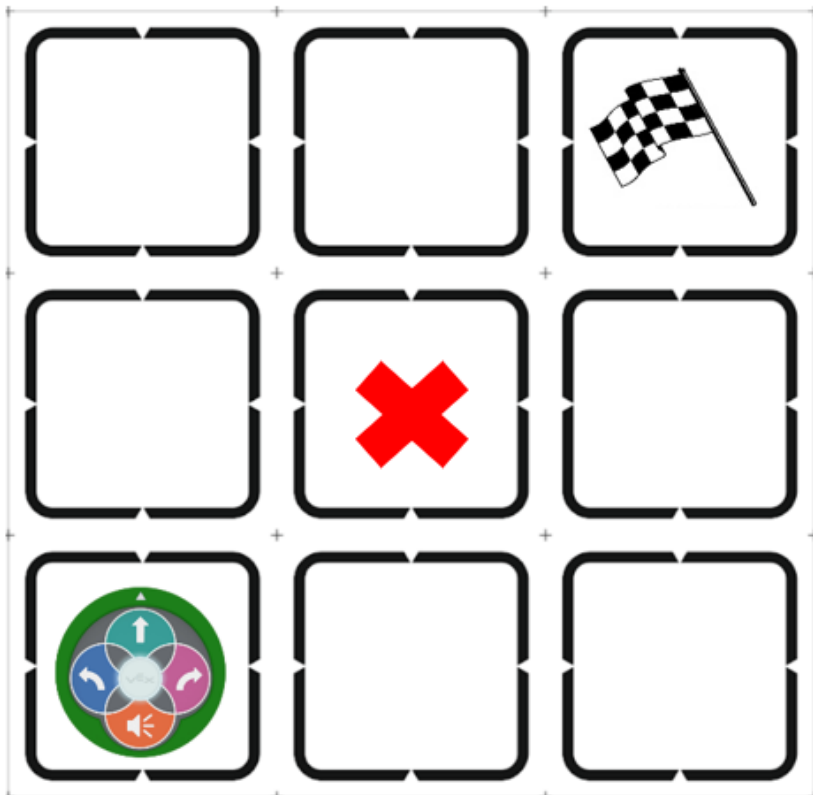
Vytvoř program, s jehož pomocí VEXík postupně dojde na cílové pole s vlajkou. Nejprve vše vyzkoušej na desce s maketou a program si zapiš do programovací tabulky. Nakonec vyzkoušej VEXíka naprogramovat a všechny úkoly splň s ním.

VEX 123 – programovací tabulka									

Jednoduché pohyby po desce - 3

- Kolik různých cest najdeš?
- V čem se jednotlivé cesty liší?
- Jde zápis programů nějak zjednodušit?
- Je nějaká cesta nejkratší (nejdelší)? Kolik jich je?

Jednoduché pohyby po desce - 4



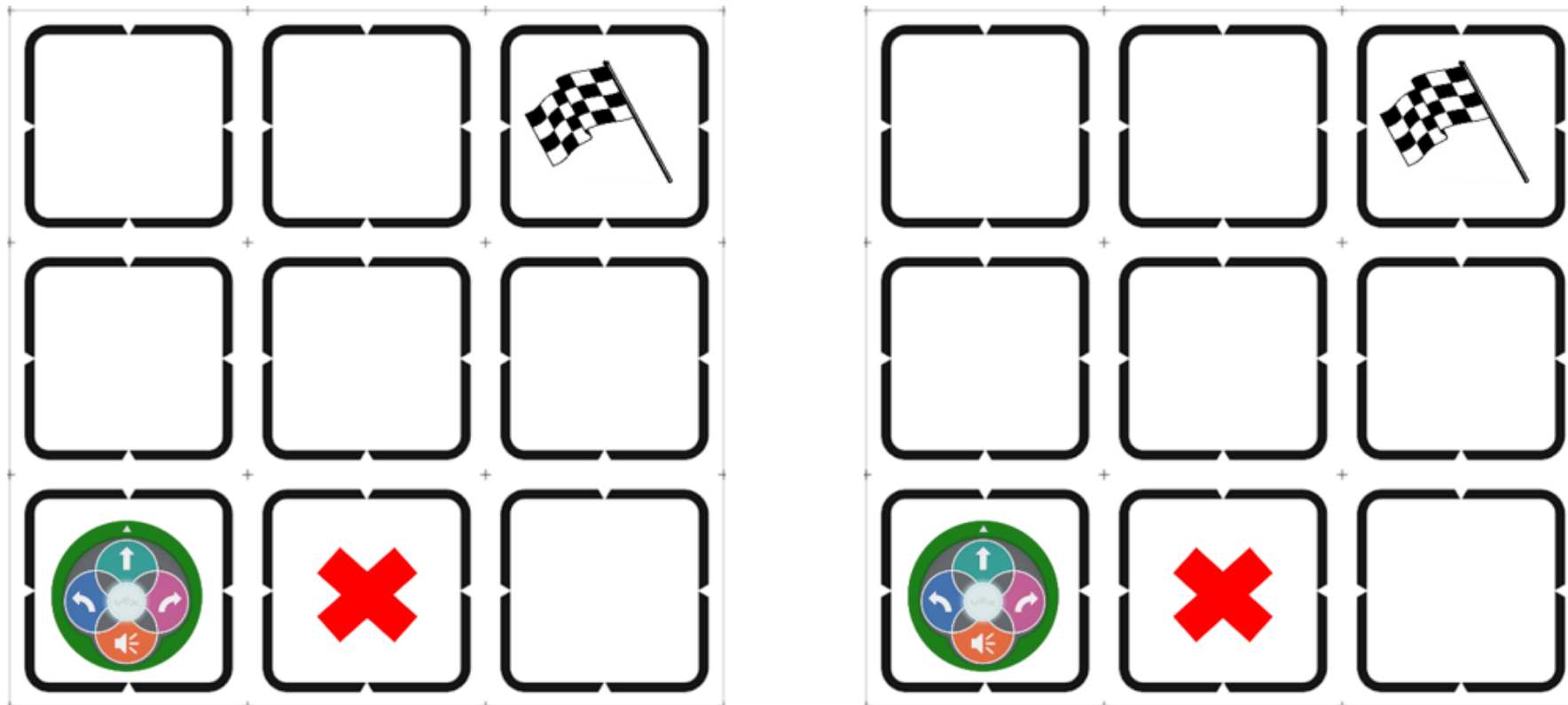
Vytvoř program, s jehož pomocí VEXík postupně dojde na cílové pole s vlajkou (červený křížek je zákaz vjezdu). Nejprve vše vyzkoušej na desce s maketou a program si zapiš do programovací tabulky. Nakonec vyzkoušej VEXíka naprogramovat a všechny úkoly splň s ním.

VEX 123 – programovací tabulka									

Metodická doporučení:

- překážku je ideální vyrobit ze dřevěných špalíčků cca 4×4×1cm ve třech barvách (červená, modrá a zelená) a bez obarvení (použitelné v dalších aktivitách na 1. i 2. stupni)
- umístěním cílové vlaječky nebo překážky lze vytvořit řadu dalších zadání

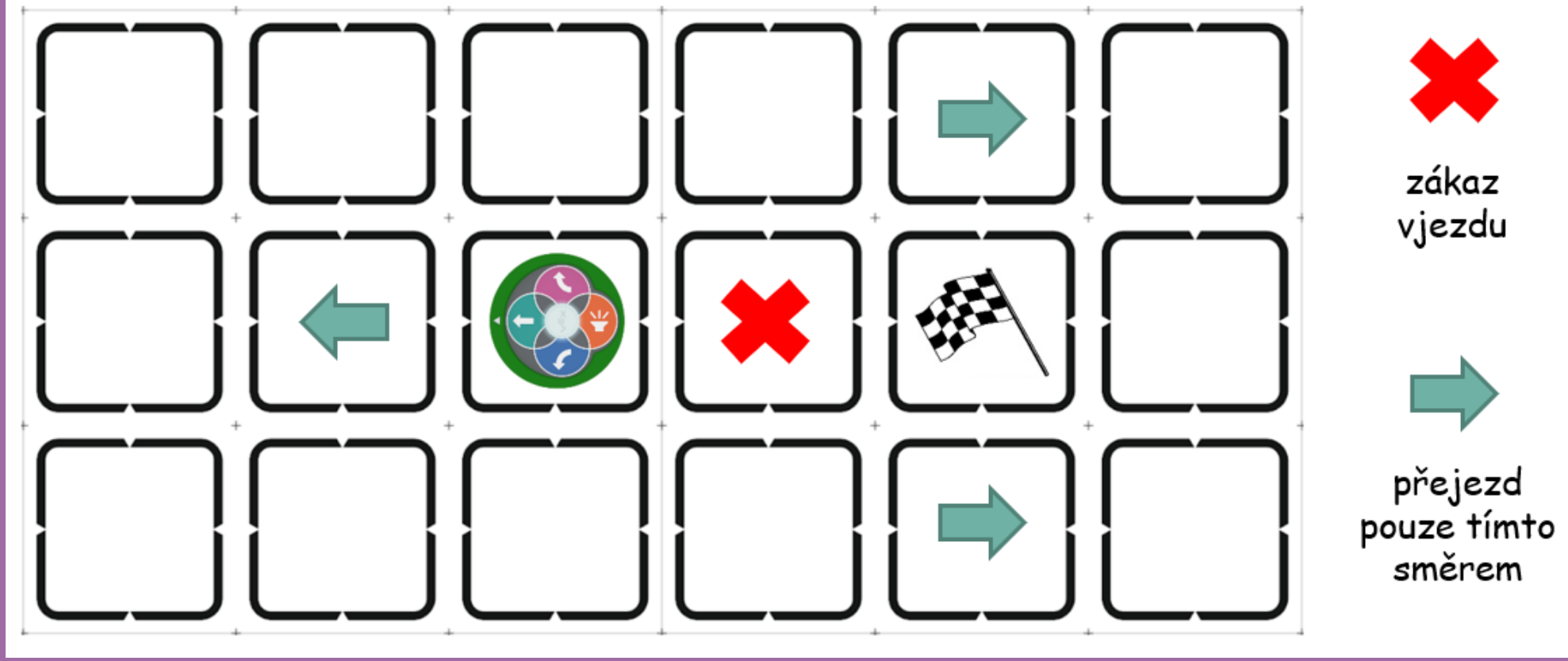
Jednoduché pohyby po desce - 5



Metodická doporučení:

- umístěním cílové vlaječky nebo překážky lze vytvořit řadu dalších zadání

Pohyb po desce - 6



Metodická doporučení:

- spojením dvou herních desek lze získat větší prostor pro pohyb VEXíka, přičemž stále stačí běžná školní lavice

Pohyb po desce - 7

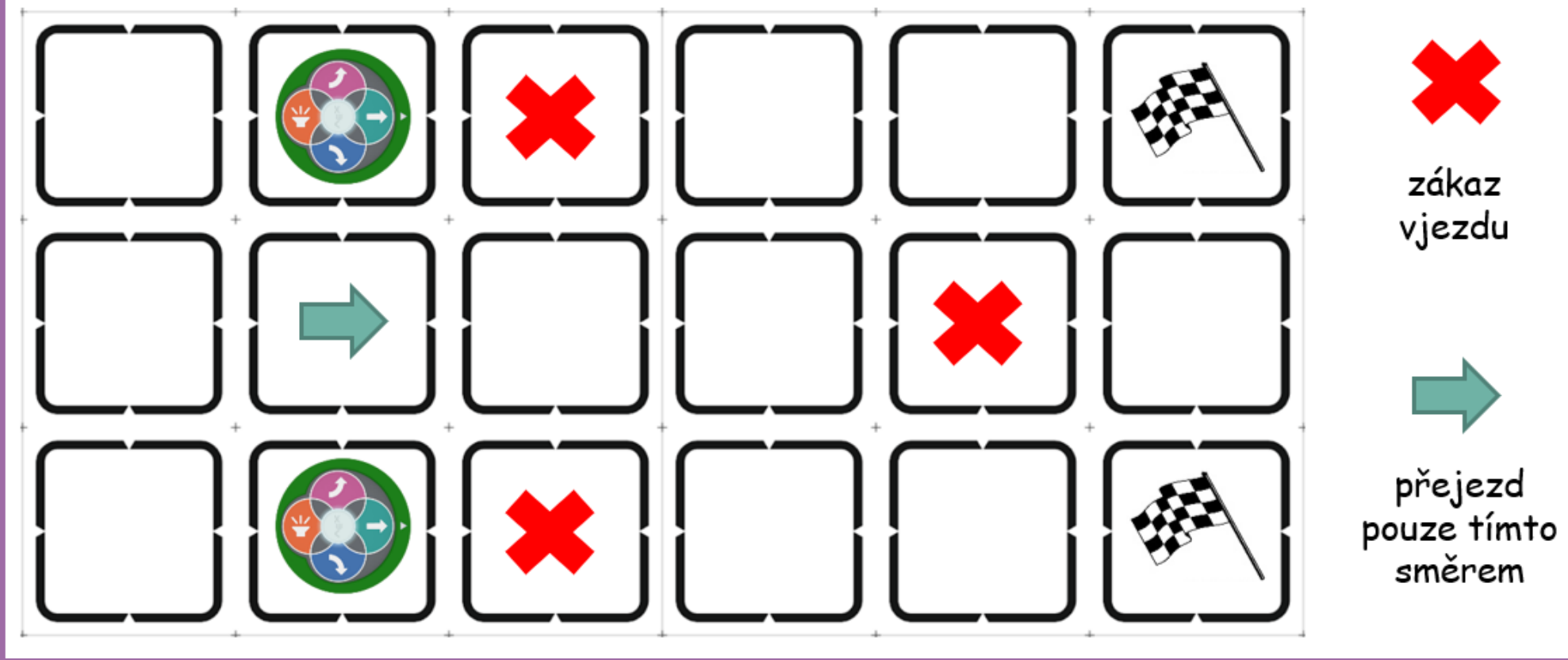
zázak vjezdu

přejezd pouze tímto směrem

Metodická doporučení:

- umístěním cílové vlaječky nebo překážky lze vytvořit řadu dalších zadání





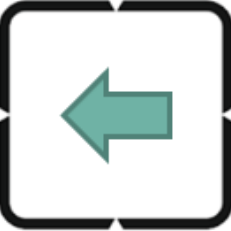



Pohyb po desce - 8 (dvojice)





Metodická doporučení:

- rozvoj plánování a spolupráce dvojic lze realizovat pohybem dvou VEXíků na jedné herní ploše
- start obou VEXíků musí být současný
- dojezd do cíle obou VEXíků může být postupný i současný

Pohyb po desce - 9 (dvojice)


zákaz vjezdu


přejezd pouze tímto směrem

Metodická doporučení:

- umístěním cílových vlaječek nebo překážek lze vytvořit řadu dalších alternativních zadání
- průjezd bludištěm lze řešit opět stejným nebo postupným dojezdem VEXíků do cíle

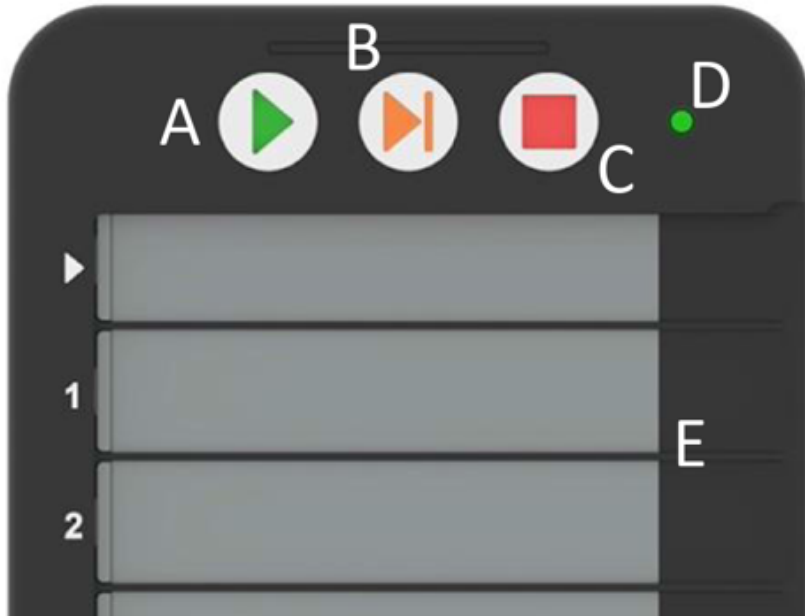
5. ročník

VEX 123

Programujeme magnetickým kóděrem



Ovládání magnetického kodéru



- A. zapnutí kodéru / spuštění programu
- B. krokové spuštění programu
- C. zastavení programu / vypnutí kodéru
- D. indikátor stavu
- E. sloty pro programovací magnetické pásky

Metodická doporučení:

- správné ovládání magnetického kodéru je základním předpokladem úspěšné tvorby programů
- výraznou pomůckou při ladění programů je krokové spuštění programu se současným sledováním LED indikátorů prováděného příkazu programu
- délka programu je omezena počtem volných slotů: 10 příkazů

Používání programovacích pásek



- slouží k vytvoření programu
- zasouvají se zprava doleva do drážek
- jsou barevně členěné do skupin
- první je vždy pásek **when start 123**

Skupiny příkazů

Pohyb

- ovládají pohyb vpřed a otáčení VEXíka

Zvuk

- ovládají zvuky vydávané VEXíkem

Vzhled

- ovládají barvu svícení středového tlačítka VEXíka

Ovládání

- řídí průběh programu a rozhodování VEXíka

Událost

- spouští nebo zastavují program

Akce

- nařizují sérii příkazů nebo akcí VEXíka

Čas

- ovládají dobu, po kterou VEXík čeká

Metodická doporučení:

- jednotlivé skupiny příkazů je vhodné nejprve žákům představit jednotlivě a předvést jejich využití na jedné desce 3×3 pole
- příkazy skupiny Akce nejsou pro výuku příliš vhodné, jsou spíše efektem bez výrazné přidané hodnoty

Otáčení

turn left



turn right



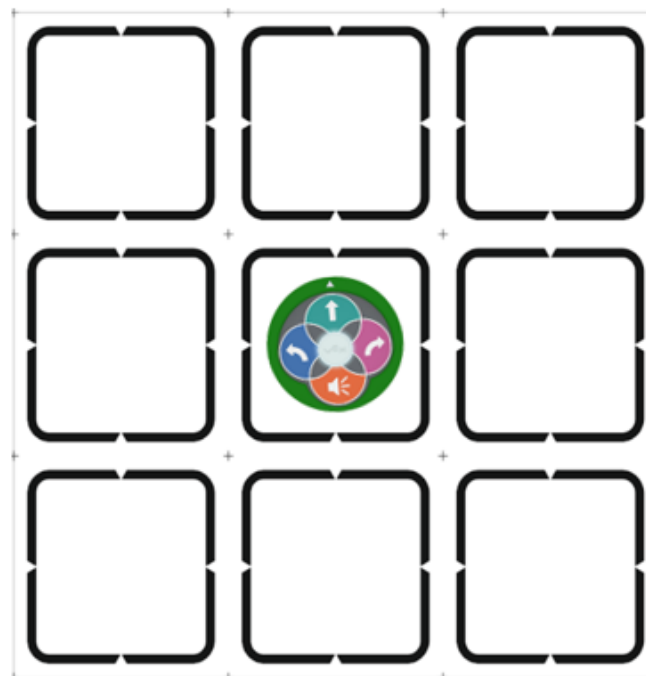
turn random



turn around



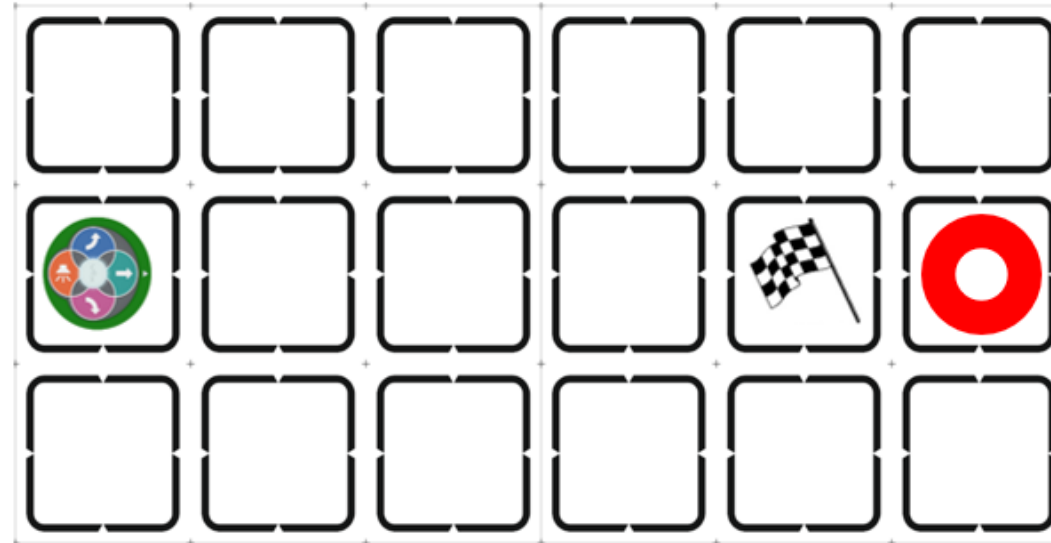
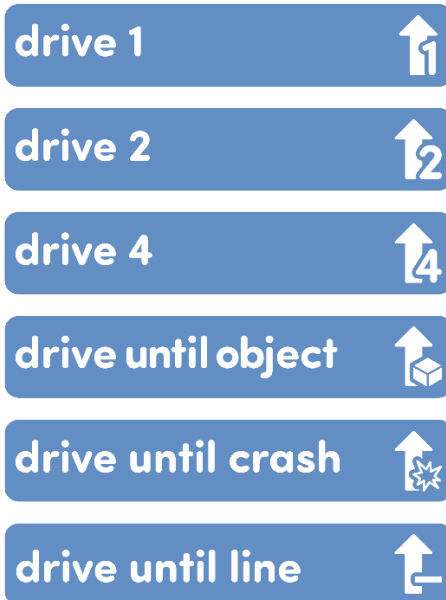
Vyzkoušej všechny způsoby,
jak se může VEXík otáčet.



Metodická doporučení:

- příkaz **turn random** je při pohybu na desce 3×3 pole velmi obtížně využitelný, alternativou může být podklad s kruhovým úhloměrem nebo „koláčem“ rozděleným na části s následným statistickým vyhodnocením četností intervalu úhlů otočení
- příkaz **turn around** provede otočení VEXíka o 180°

Pohyb



Naprogramuj VEXíka tak, aby dojel na cílové pole. Porovnej VEXíkovu činnost při použití různých příkazových pásek.

Metodická doporučení:

- žáci se mají naučit používat vícenásobné pohyby dopředu
- byť v učebnici Scratch pro 5. ročník není využito příkazů s podmínkou (rozhodování nebo cyklus), tak poprvé žáci intuitivně vyzkouší využívání optického čidla (příkaz **drive until line** nesleduje čáru jako např. Ozobot, jen ji registruje)
- zastavovací překážkou může být dodaný barevný plastový kroužek nebo barevný dřevěný (plastový) hranolek 4×4×1 cm

Zvuk, vzhled

play honk



play doorbell



play crash



play random



glow purple



glow green



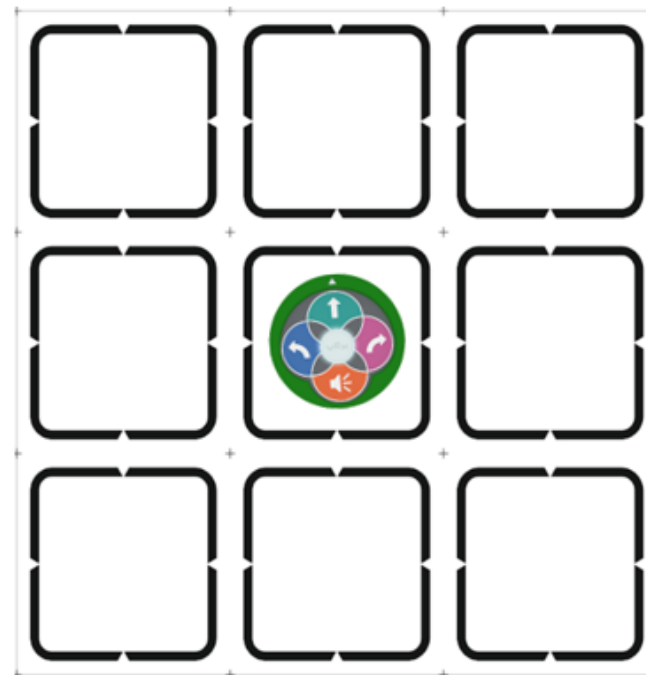
glow blue



glow off



Vyzkoušej všechny způsoby, jak může VEXík vydávat zvuky a svítit středovým tlačítkem.



Metodická doporučení:

- žáci se mají naučit používat zvuku a svícení středového tlačítka
- postupně lze kombinovat programování zvuku a vzhledu s pohybovými příkazy (popojed' a rozsvít, zadrž atd.)
- kombinace příkazů vzhledu, zvuku a pohybu lze praktikovat také na větším hřišti spojením dvou herních polí

Ovládání 1

if right button 

if left button 

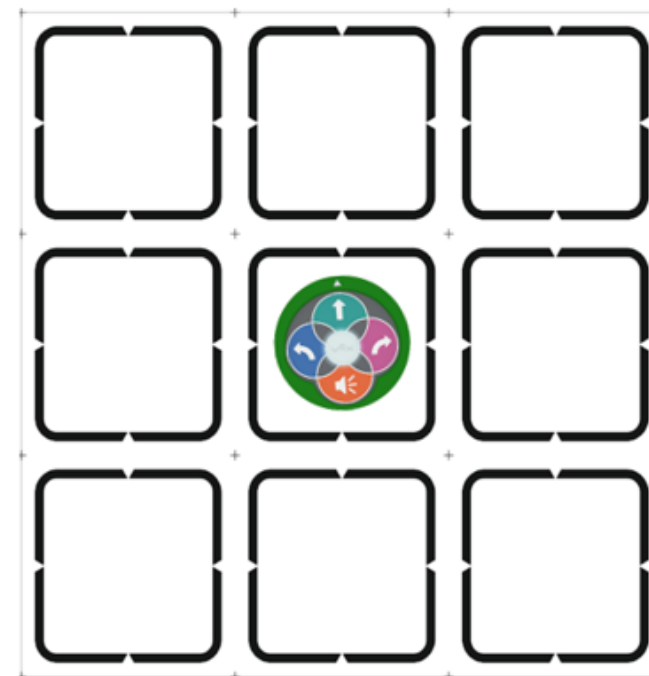
if move button 

if sound button 

if red 

if green 

if blue 



Vyzkoušej reakci VEXíka:

- na stisk tlačítka provede akci tlačítka
- podle detekované barvy začne stejnou barvou blikat středovým tlačítkem

Metodická doporučení:

- žáci se mají naučit využívání reakcí na události, které je schopen VEXík registrovat (stisk tlačítka a optický senzor)
- postupně lze kombinovat s programováním zvuku, vzhledu a pohybovými příkazy (popojed' a rozsviť, zatrub atd.)
- vyhodnocovacím předmětem může být dodaný barevný plastový kroužek nebo barevný dřevěný (plastový) hranolek 4×4×1 cm
- pro správnou činnost vyhodnocení událostí je zapotřebí program uvést do nekonečného cyklu příkazem **goto start**

Ovládání 2

if object



if dark



if no object



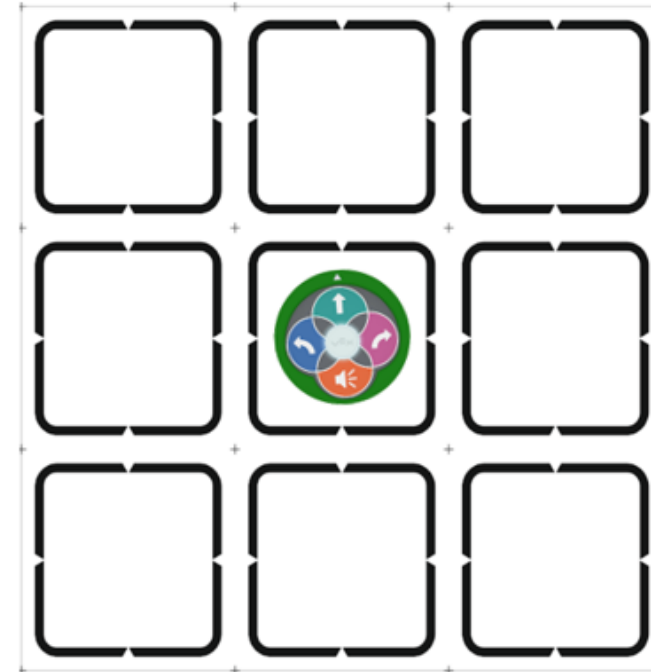
else

if crash



end if

if bright



Vyzkoušej reakci VEXíka:

- na přítomnost objektu
 - na tmou a světlo
- ukončení a větvení podmínky

Metodická doporučení:

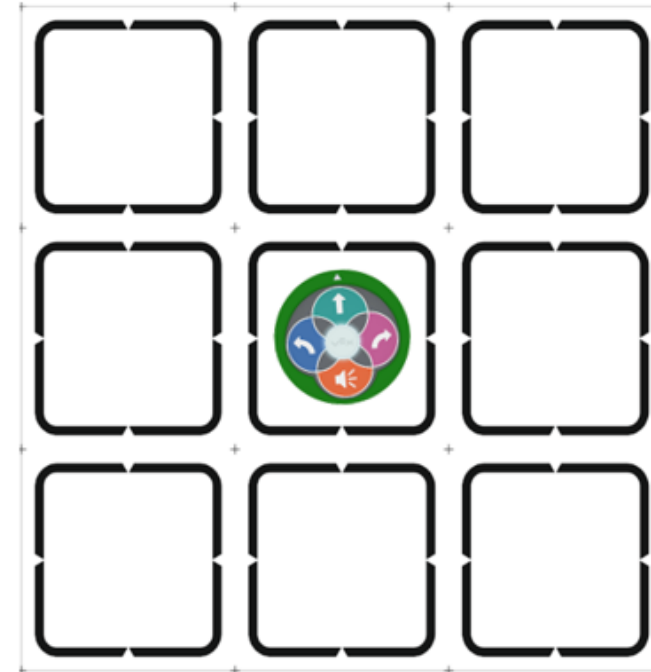
- žáci se mají naučit používání podmíněného příkazu **if** (když) a jeho rozšíření **else** (jinak) s použitím optického senzoru
- postupně lze kombinovat s programováním zvuku, vzhledu a pohybovými příkazy (popojed' a rozsviť, zatrub atd.)
- vyhodnocovacím předmětem může být dodaný barevný plastový kroužek nebo barevný dřevěný (plastový) hranolek 4×4×1 cm
- lze vytvářet úlohy typu „když je objekt, vydej zvuk, blikni“ apod.

Akce, čas a událost



Vyzkoušej VEXovi:

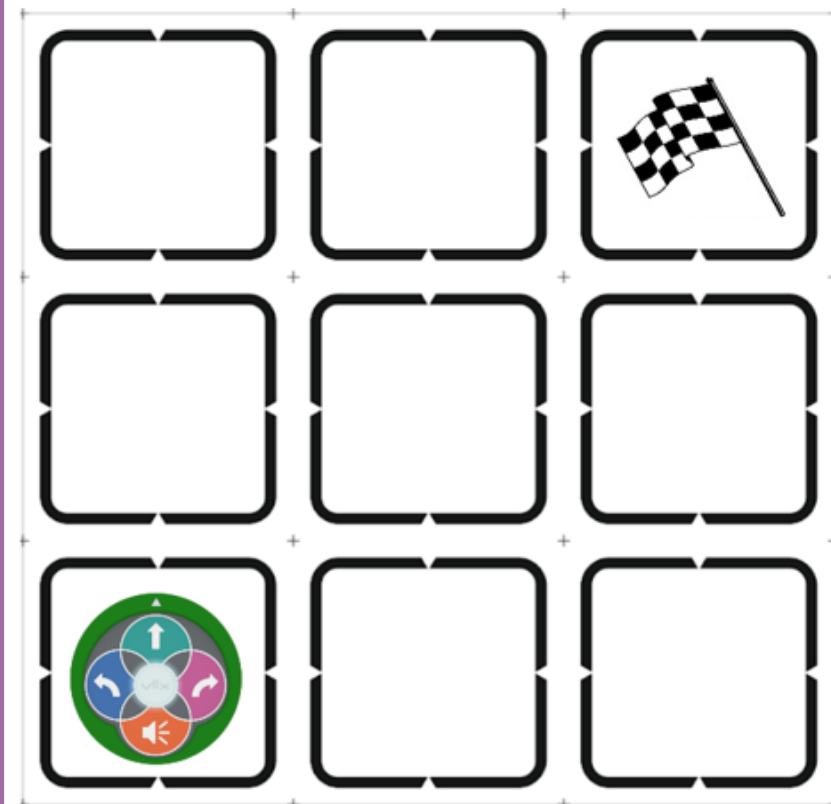
- jednotlivé akce
- čekání po určitou dobu
- zastavení a vrácení programu



Metodická doporučení:

- žáci si vyzkouší předprogramované VEXíkovy akce (buď šťastný, bláznivý, smutný) a příkazy čekání
- postupně lze kombinovat s programováním zvuku, vzhledu a pohybovými příkazy (popojed' a rozsviť, zatrub atd.)
- tyto akce nemají pro potřeby výuky algoritmizace příliš velký význam, jedná se pouze o zpestření pohybových možností

Jednoduché pohyby po desce - 1



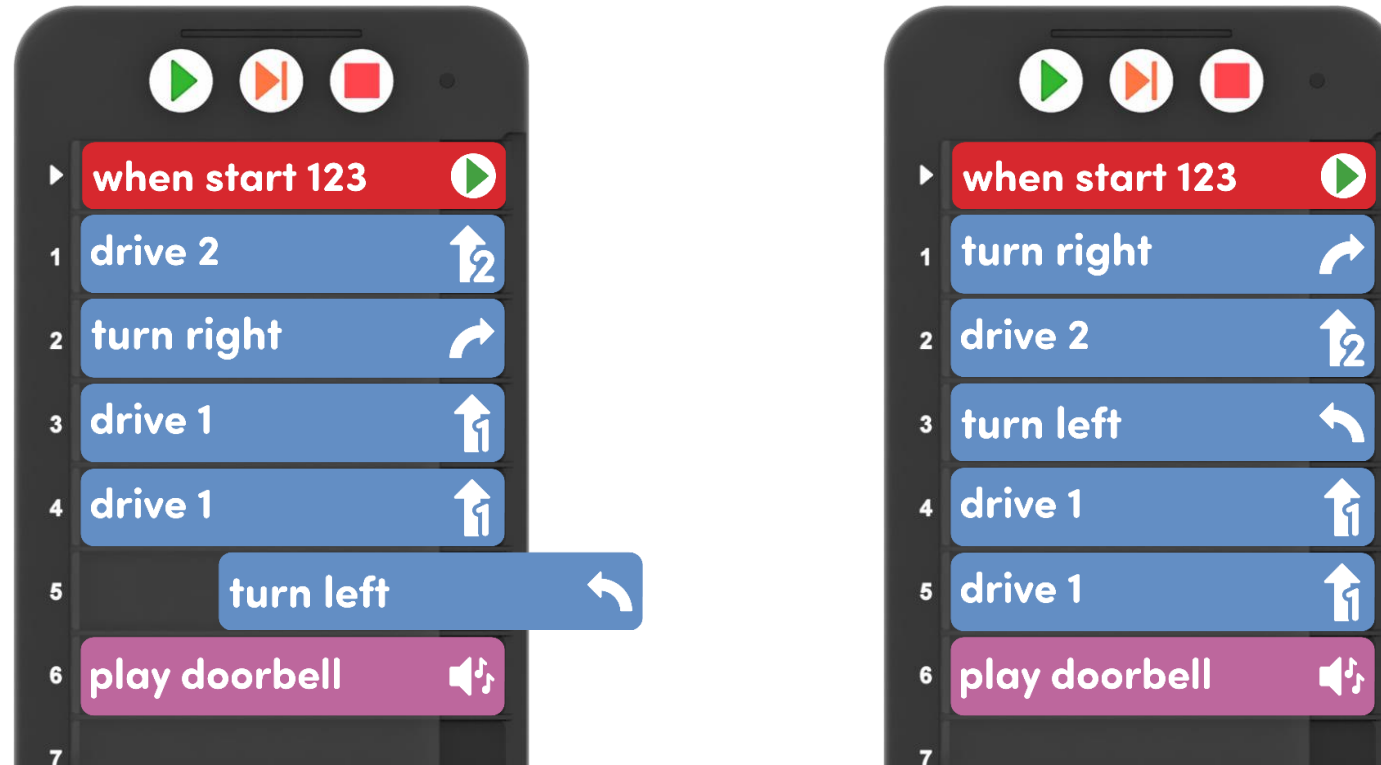
Vytvoř program, s jehož pomocí VEXík postupně dojede na cílové pole s vlajkou. Nejprve vše vyzkoušej na desce s maketou a program sestav v kodéru.

- Jak se budou lišit cesty po obvodu a napříč deskou?
- Doplň program vhodně tak, aby se VEXík dostal zase zpět na výchozí pole.

Metodická doporučení:

- cílem této aktivity je tvorba **sekvencí**, které budou používat pouze pohybové příkazy
- délka programu a tím také délka pohybu jsou omezeny počtem slotů v programátoru
- alternativní aktivity lze zadávat různým rozmístěním výchozího a cílové pole

Jednoduché pohyby po desce - 1a



Metodická doporučení:

- existují 2 nejkratší cesty, z nichž jedna může být na základě požadavku cílové orientace o 1 příkaz kratší

Jednoduché pohyby po desce - 1b



Metodická doporučení:

- řešení obsahuje cesty napříč středovou řadou polí (vodorovně, svisle)

Jednoduché pohyby po desce - 1c



Metodická doporučení:

- schodovitá řešení cesty na cílové pole jsou v obou případech stejná co do počtu polí i příkazů

Jednoduché pohyby po desce - 1d



Metodická doporučení:

- řešení využívá optického senzoru VEXíka, je nutné využít mantinelů na stranách, kde dojde k otočení VEXíka

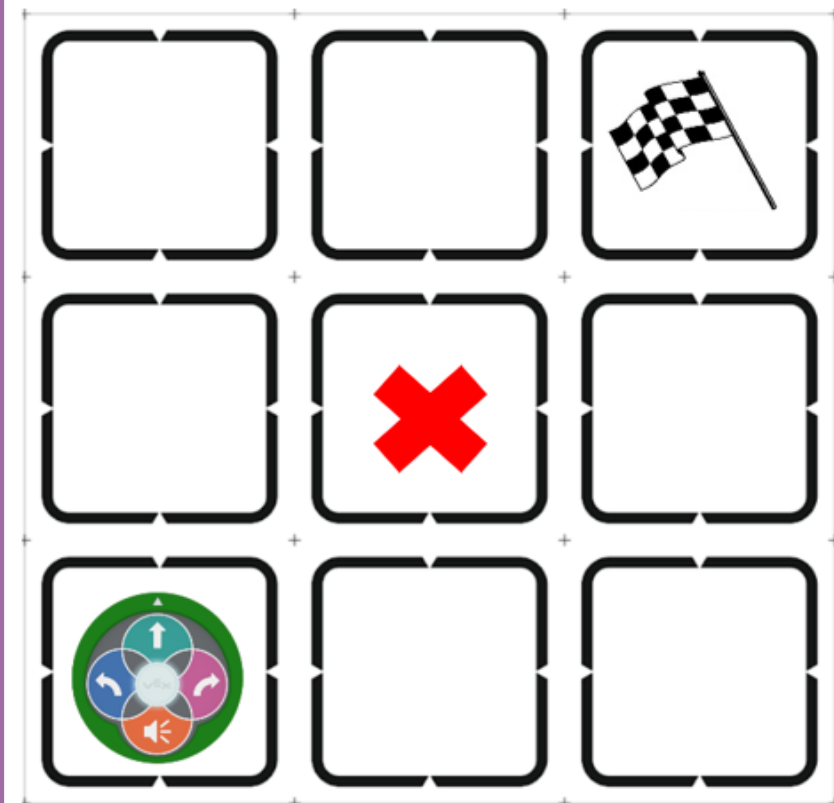
Jednoduché pohyby po desce - 1 - otázky

1. Která cesta je nejkratší podle počtu navštívených čtverců?
2. Která cesta je nejkratší podle počtu použitých blokových příkazů?
3. Která cesta je nejdelší podle počtu navštívených čtverců?
4. Která cesta je nejdelší podle počtu použitých blokových příkazů?
5. Kterou cestu nelze realizovat pomocí magnetického programátoru?

Metodická doporučení:

- je vhodné žákům promítnout úvodní zadání a nechat je hledat a objevovat různé varianty řešení
- aktivitu je vhodné zakončit diskuzí o objevených řešeních a jejich tříděním do skupin

Jednoduché pohyby po desce - 2

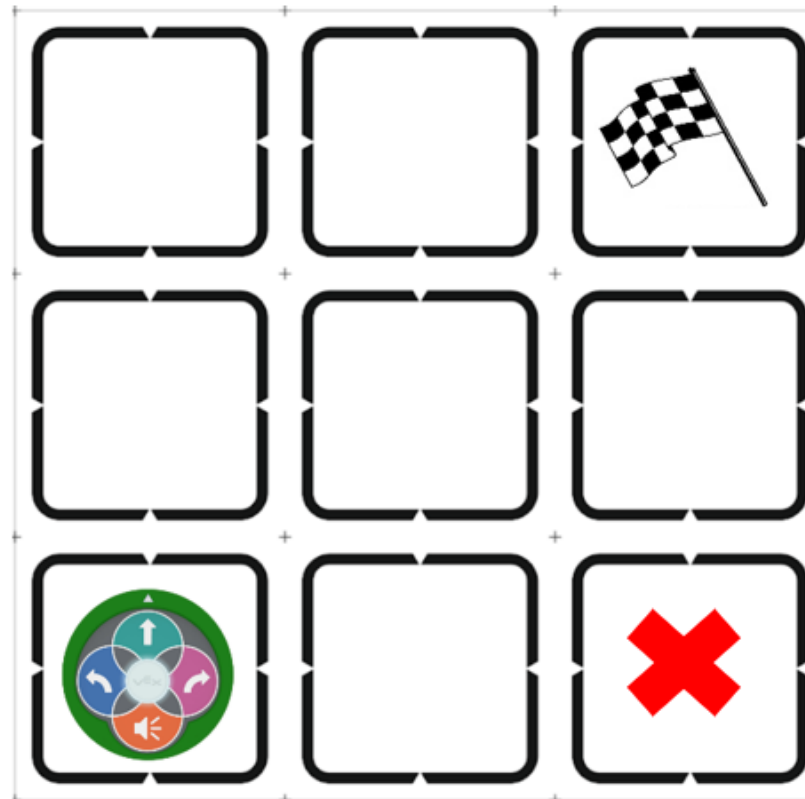
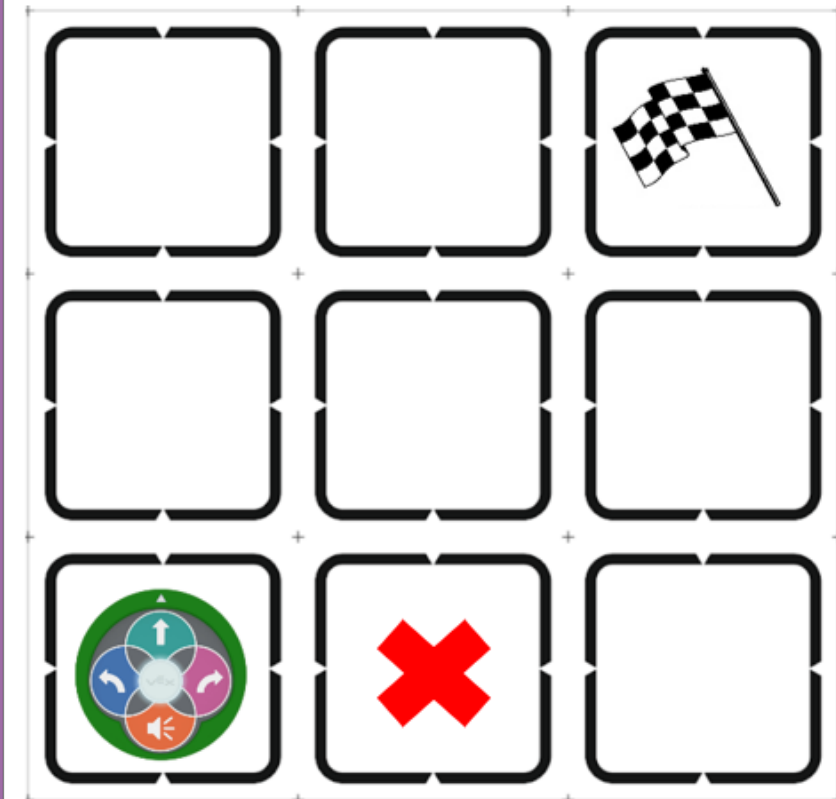


Vytvoř program, s jehož pomocí VEXík postupně dojde na cílové pole s vlajkou (červený křížek je zákaz vjezdu). Nejprve vše vyzkoušej na desce s maketou a program sestav v kodéru. Nakonec vyzkoušej VEXíka naprogramovat a všechny úkoly splň s ním.

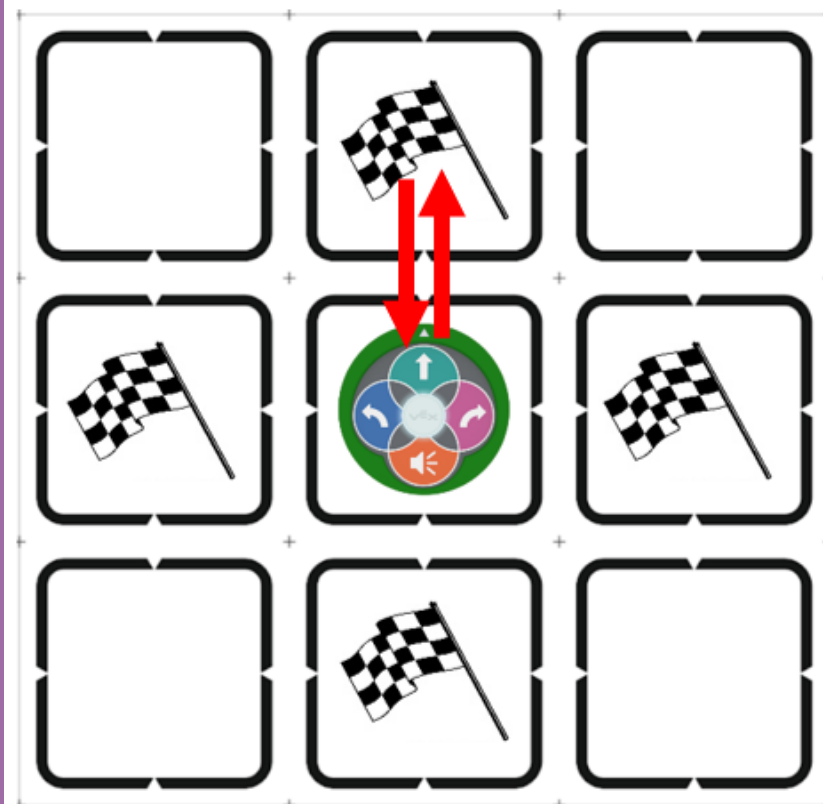
Metodická doporučení:

- obdobné aktivity lze realizovat a podobně diskutovat nalezená řešení různým umístěním překážky, startovací a cílové polohy VEXíka, přičemž je nutné dbát využívání pouze dostupného počtu příkazů v každé sadě programovacích proužků

Jednoduché pohyby po desce - 3



Jednoduché pohyby po desce - 4



Vytvoř program, s jehož pomocí VEXík postupně dojede na cílová pole s vlajkou a zpět. Nejprve vše vyzkoušej na desce s maketou a program pomocí programovacích pásků seskládej do programátoru. Nakonec vyzkoušej VEXíka naprogramovat pomocí programátoru a všechny úkoly splň s ním.

Dokážeš navrhnout program tak, aby VEXík opakovaným spouštěním programu objel postupně všechna cílová pole?

Metodická doporučení:

- žáci nejprve řeší problém pohybu na jednotlivá cílová pole (4 úlohy)
- na základě řešení všech 4 pohybů se snaží zobecnit řešení a využít cyklování prvního řešení s následujícím otočením

Jednoduché pohyby po desce - 4



Jednoduché pohyby po desce - 5



Vytvoř program, s jehož pomocí VEXík postupně dojde na cílová pole s vlajkou a zpět. Nejprve vše vyzkoušej na desce s maketou a program pomocí programovacích pásků seskládej do programátoru. Nakonec vyzkoušej VEXíka naprogramovat pomocí programátoru a všechny úkoly splň s ním.

Dokážeš navrhnout program tak, aby VEXík opakovaným spouštěním programu objel postupně všechny cílová pole?

Jednoduché pohyby po desce - 5



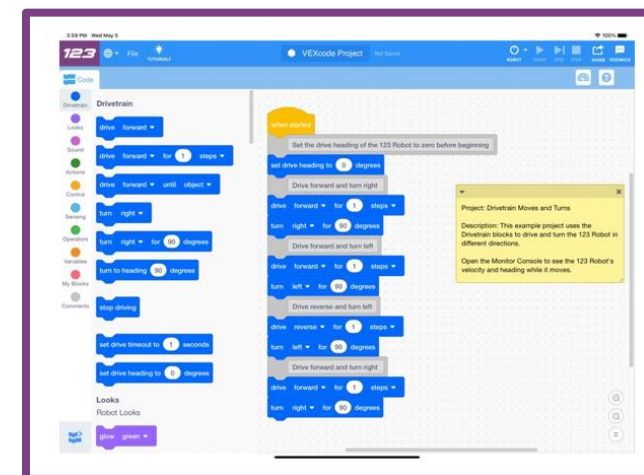
Metodická doporučení:

- vzhledem k 10 volným slotům v kodéru nelze využít cyklování programu, ale jen opakovaného spuštění
- žáci by měli ve skupinách zjistit a porovnat řešení „ve směru“ a „proti směru hodinových ručiček“

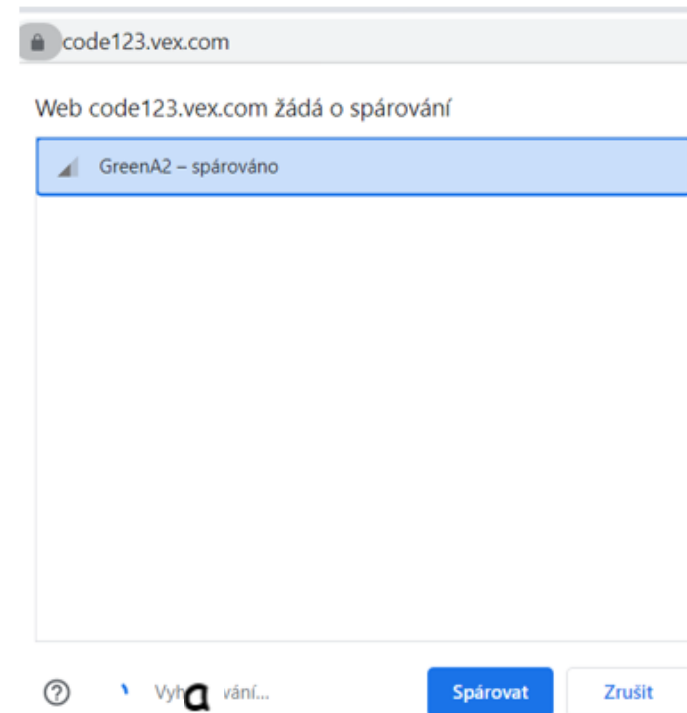
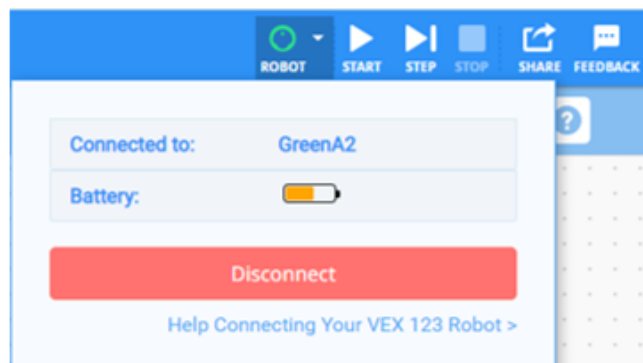
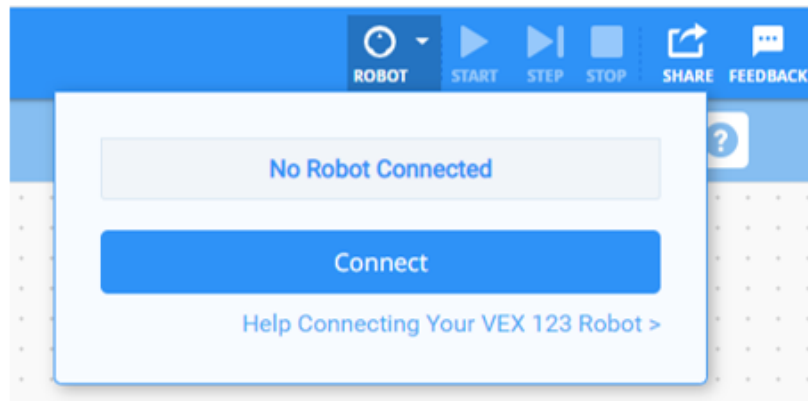
6. ročník

VEX 123

Programování blokovými příkazy



Spárování VEX 123 a VEX 123 Code



Metodická doporučení:

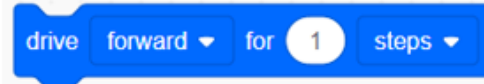
- spárování VEXíka s počítačem pomocí Bluetooth lze provést bez problémů v libovolném prohlížeči
- kvůli úspoře energie se VEXík standardně po 5 minutách vypíná, je nutné, aby žáci spárování zvládali zcela automaticky
- VEXíky je nutné vhodně označit, aby žáci nepárovali s počítačem jiného než svého VEXíka

Příkazy pohybu (experiment na dvojdesce)



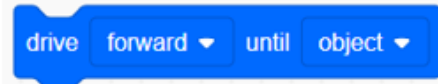
drive forward ▾

JEĎ vpřed/vzad



drive forward ▾ for 1 steps ▾

JEĎ vpřed/vzad o **X** kroků/mm



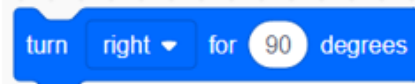
drive forward ▾ until object ▾

JEĎ vpřed/vzad dokud není **OBJEKT/SRÁŽKA/ČÁRA**



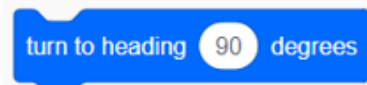
turn right ▾

DOPRAVA/DOLEVA



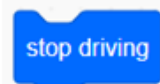
turn right ▾ for 90 degrees

DOPRAVA/DOLEVA o **X** stupňů



turn to heading 90 degrees

OTOČ SE o **X** stupňů



stop driving

ZASTAV

Metodická doporučení:

- příkazy pohybu je vhodné se žáky vyzkoušet před sestavováním delších programů, ideálně na 2 spojených deskách
- pro spuštěné programy nedoporučuji ve větší míře používat „mantinely“, VEXík o ně občas zadrhne, vhodné jen jako konec dráhy pohybu
- příkazy JEĎ vpřed/vzad a DOPRAVA/DOLEVA nemají smysl používat, jedná se o tzv. „nekonečné“ pohyby bez zastavení

Příkazy reakcí (experiment VEX 123 a VEX code)

glow green

SVIŤ zelená/modrá/fialová/zhasni

play sound honk

ZAHRAJ honk / crash / doorbell / náhodně

act sad

AKCE smutný / šťastný / šílený

print VEXcode

VYTISKNI text / a odřádkuj

set print precision to 0.1

NASTAV PŘESNOST TISKU na X

clear all rows

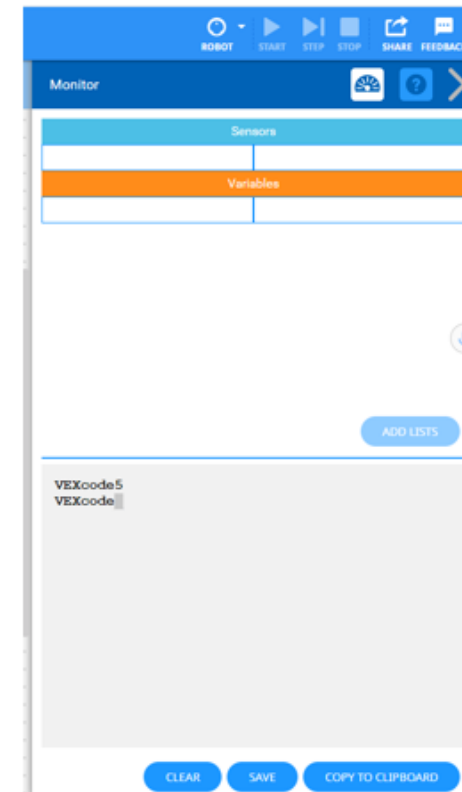
SMAŽ VŠECHNY ŘÁDKY

set cursor to next row

ODŘÁDKUJ

set print color black

NASTAV BARVU TISKU barva



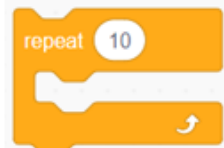
Metodická doporučení:

- pro potřeby výuky v 6. ročníku je vhodné používat pouze příkazy pohybu, vzhledu a zvuku
- je vhodné již v 6. ročníku začít pracovat s dialogovým panelem VEXíka na počítači pro základní komunikaci (výpis), jedná se o v budoucnu velmi šikovně používanou možnost práce s robotem

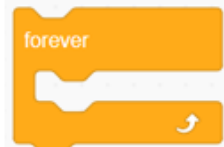
Příkazy pro ovládání programu



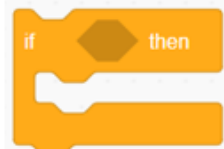
ČEKEJ X SEKUND



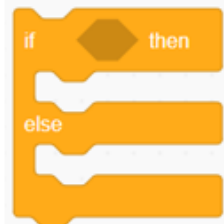
OPAKUJ X krát



NEUSTÁLE DĚLEJ



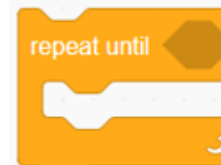
**KDYŽ je splněná
PODMÍNKA**



**KDYŽ je splněná
PODMÍNKA JINAK**



**ČEKEJ DO SPLNĚNÍ
PODMÍNKY**



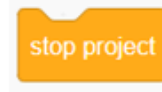
**OPAKUJ DO SPLNĚNÍ
PODMÍNKY**



**DOKUD JE SPLNĚNÁ
PODMÍNKA OPAKUJ**



PŘERUŠIT

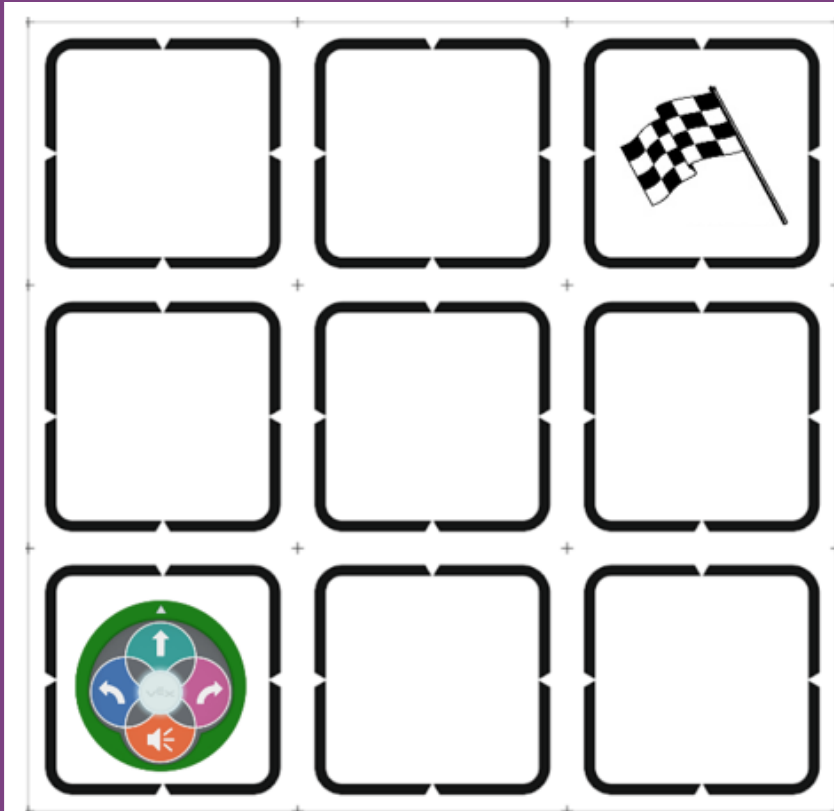


ZASTAV PROJEKT

Metodická doporučení:

- pro potřeby výuky v 6. ročníku stačí využívat pouze příkazy ČEKEJ, OPAKUJ X krát a NEUSTÁLE DĚLEJ
- příkaz ZASTAV PROJEKT je nutný pro správné ukončení programu a VEXíkovy činnosti

Jednoduché pohyby po desce - 1



Vytvoř program, s jehož pomocí VEXík postupně dojde na cílová pole s vlajkou a zpět. Nejprve vše vyzkoušej na desce s maketou a program zapiš pomocí blokových příkazů do jednoho scénáře. Nakonec vyzkoušej VEXíka naprogramovat a všechny úkoly splň s ním.

Metodická doporučení:

- stejné úlohy byly řešeny již dříve, jde pouze o úvodní seznámení s blokovými příkazy a jejich správným používáním při sestavování sekvencí, lze využít také zadání s překážkami a stejnými otázkami

Jednoduché pohyby po desce - 1

```
when started
clear all rows
print "Ahoj, právě startuji!" ◀ and set cursor to next row
glow green ▼
drive forward ▼ for 2 steps ▼
print "Odbočuj doprava!" ◀ and set cursor to next row
wait 1 seconds
glow purple ▼
```

```
turn right ▼ for 90 degrees
drive forward ▼ for 2 steps ▼
print "Ještě se musím otočit do původního směru!" ◀ and set cursor to next row
turn left ▼ for 90 degrees
print "Jsem v cíli!" ▶
glow green ▼
play sound doorbell ▼
glow off ▼
stop project
```

Pohyby po desce - 2

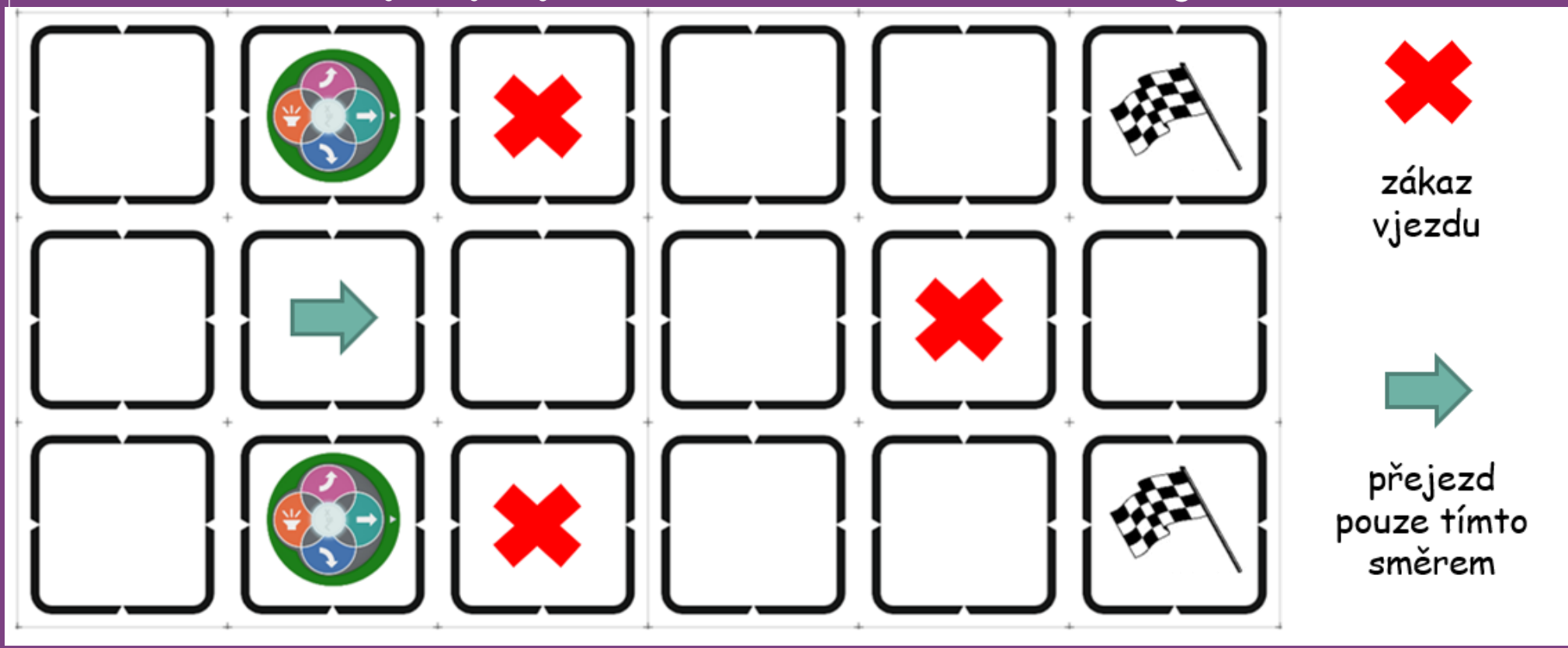
zákaz vjezdu

přejezd pouze tímto směrem

Metodická doporučení:

- spojením dvou herních desek a přidáním překážek lze připravovat různá prostředí, ve kterých žáci musí naprogramovat správný průjezd ze startovacího pole do cílového
- lze využít i spojení většího počtu desek pro tvorbu rozlehlejších prostředí, zajímavé je zapojení žáků do tvorby











Pohyby po desce - 3 (dvojice)



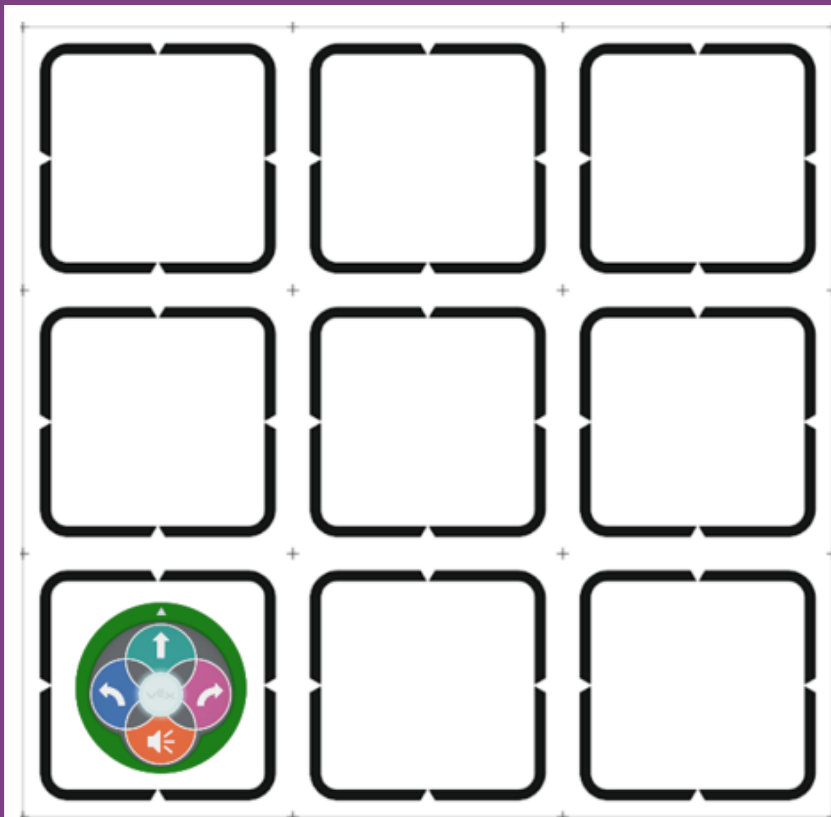
Metodická doporučení:

- zajímavou výzvou pro žáky může být opět tvorba programů pro dva a více VEXíků v rozlehlejších prostředích, kdy je cílem projetí správné cesty ze startu do cíle pro více VEXíků při současném startu
- lze využít i spojení většího počtu desek pro tvorbu rozlehlejších prostředí, zajímavé je zapojení žáků do tvorby

Pohyby po desce - 4 (dvojice)

							zákaz vjezdu
							přejezd pouze tímto směrem
							

Jednoduché cykly - 1



Vytvoř program, kterým VEXík na desce objede:

- řadu čtvercových polí
- malý čtverec 2x2 pole
- velký čtverec 3x3 pole
- obdélník 2x3 pole
- obdélník 3x2 pole

Porovnej použití cyklu pro jízdu s číselným zadáním délky pohybu.

Metodická doporučení:

- seznámení s jednoduchými cykly OPAKUJ X KRÁT (**repeat**) a NEUSTÁLE DĚLEJ (**forever**) navazuje na používání podobných příkazů v prostředí Scratch

```
when started
repeat 2
  drive forward for 1 steps
stop project
```

```
when started
repeat 4
  drive forward for 1 steps
  turn right for 90 degrees
stop project
```

```
when started
repeat 2
  drive forward for 1 steps
  turn right for 90 degrees
stop project
```

Propojením oblastí **Algorithmizace a Data, informace** může být zařazení úlohy na vytvoření programu, kdy VEXík bude generovat sérii náhodných čísel v určitém celočíselném intervalu (např. 1-6 jako simulace házení hrací kostkou). Žáci pak tuto skupinu náhodných čísel mohou vizualizovat pomocí online nástroje <https://www.chartgo.com/index.jsp> ve své pracovní dvojici a po vytvoření třídního souboru dat pak i v rozsahu celé třídy a porovnávat výsledky náhodného generování (pravděpodobnost a statistika):

The image shows a Scratch script on the left and a data table on the right. The script is as follows:

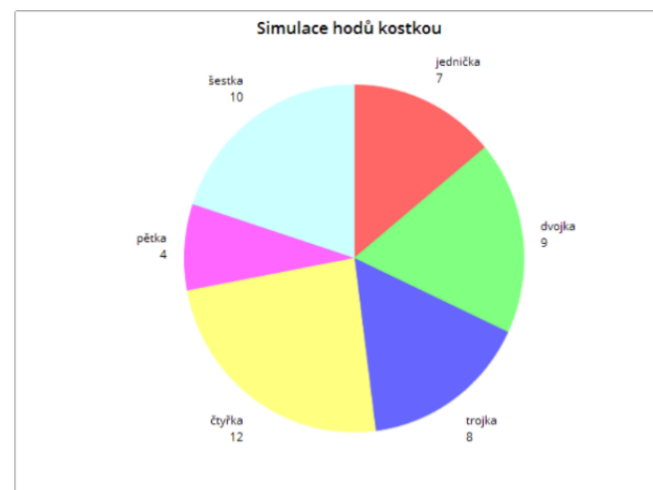
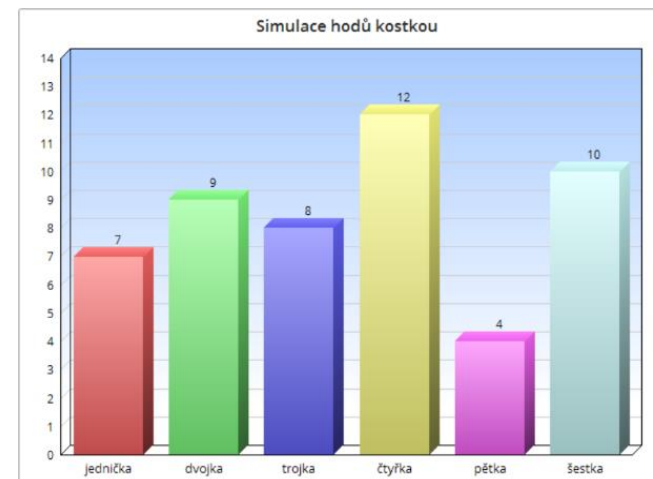
```

when started
  repeat 5
    repeat 10
      print pick random 1 to 6
      print ,
      play sound honk
      wait 1 seconds
    print and set cursor to next row
  
```

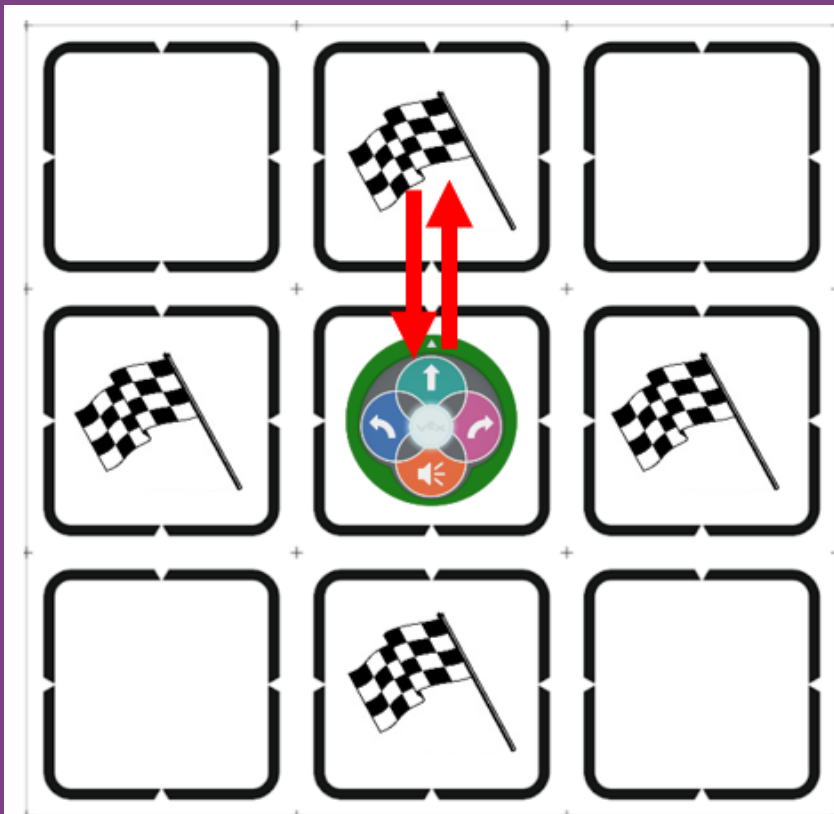
The data table on the right is titled "Sensors" and "Variables". Below it, a list of 50 random numbers is displayed:

```

3, 6, 1, 4, 5, 4, 6, 2, 6, 6,
4, 4, 4, 3, 6, 4, 4, 1, 1, 6,
4, 3, 6, 3, 4, 2, 5, 6, 1, 4,
4, 5, 5, 6, 1, 1, 4, 2, 2, 3,
2, 2, 1, 2, 2, 3, 3, 6, 3, 2,
  
```



Vlastní bloky - 1



Vytvoř program, s jehož pomocí VEXík postupně dojede na cílová pole s vlajkou a zpět. Nejprve vše vyzkoušej na desce s maketou a program zapiš pomocí blokových příkazů do jednoho scénáře. Nakonec vyzkoušej VEXíka naprogramovat a všechny úkoly splň s ním.

Metodická doporučení:

- aplikace jednoduchých cyklů a především používání vlastních bloků za účelem zjednodušení a zefektivnění programu je mimořádně důležité pro další roky výuky algoritmizace – zvládnutí této úlohy se žáky je proto zcela zásadní

Realizace výukové aktivity

Cílová skupina žáků: 6. ročník ZŠ

Předpoklady: žáci v rámci výuky Informatiky již absolvovali a zvládli z oficiálních online učebnic MŠMT:

- tvorba jednoduchých scénářů
- tvorba vlastních bloků

Cíl aktivity:

- rozklad problému na dílčí části
- vytvoření řídicího a dílčích scénářů

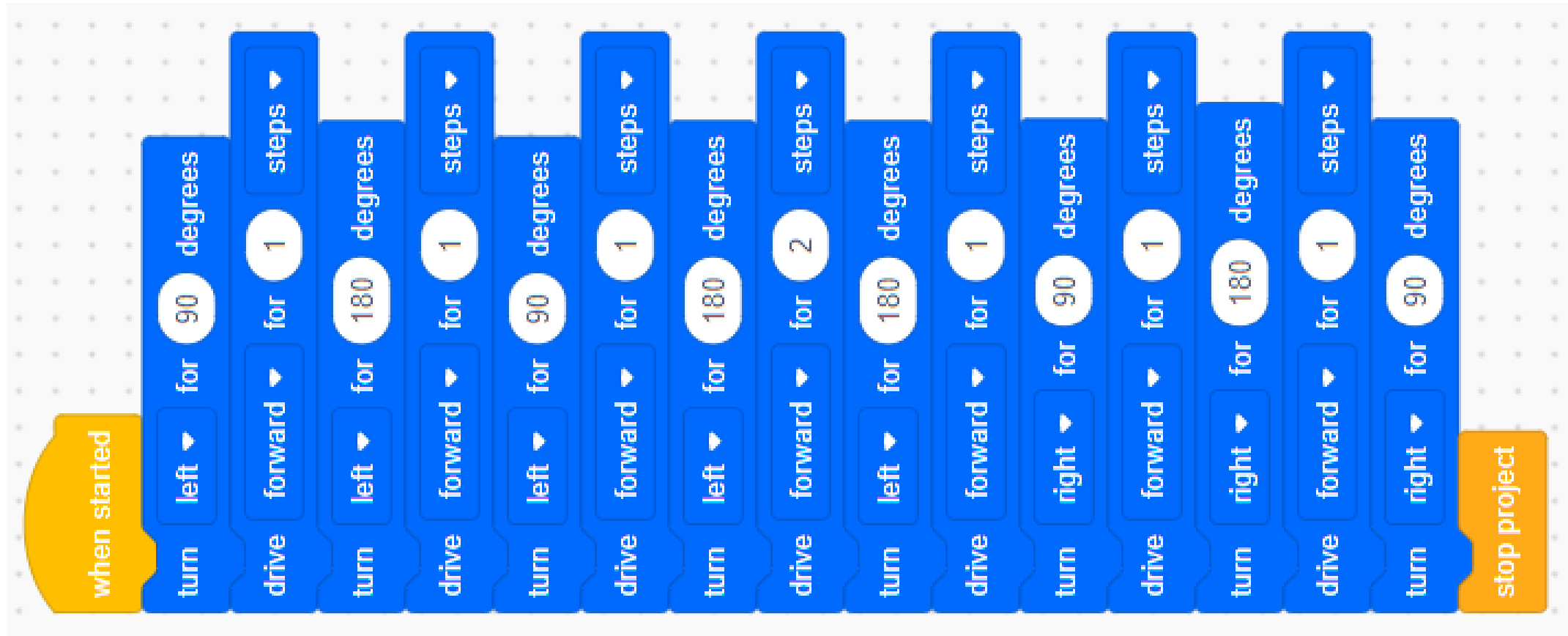
Zadaní úlohy:

Vytvořte scénář, podle kterého VEXík projede vlajkami v pořadí LEVÁ – HORNÍ – DOLNÍ – PRAVÁ. Vytvořený scénář by měl být co nejsrozumitelnější.

Postup realizace:

1. žáci vytvoří vlastní řešení pomocí dlouhého a nepřehledného scénáře bez použití vlastních bloků
2. žáky vedeme k dalšímu zefektivnění a zpřehlednění scénáře s využitím vlastních bloků (větší množství krátkých scénářů je vždy efektivnější a přehlednější než jediný dlouhý a nepřehledný scénář)
3. efektivnost tohoto postupu žákům předvedeme změnou pořadí navštívených cílových pozic s upozorněním na složitost tohoto úkolu bez použití vlastních bloků
4. použití vlastních bloků sice vytváří větší počet použitých příkazových bloků, je ale nakonec mnohem přehlednější

Ukázka řešení ve formě dlouhého scénáře bez vlastních bloků:



Metodická doporučení:

- jednotlivé dvojice žáků mohou sestavit mírně odlišná řešení v závislosti na směru otáčení VEXíka a počtu kroků nebo úhlu otočení robota – správná jsou veškerá řešení, kterými VEXík splní zadaný úkol – je třeba následně poukázat na jejich nepřehlednost a složitou variabilitu přeskupováním jednotlivých bloků – snažíme se najít „společné části řešení“ = vlastní bloky

Ukázka řešení s využitím stejných částí scénáře – pohyb na vlajku a zpět na start:

The image displays a Scratch script on a grid background. At the top left is a yellow 'when started' block. Below it are four columns of code blocks, each representing a different direction: LEVÁ, HORNÍ, DOLNÍ, and PRAVÁ. Each column is introduced by a grey comment block with the direction name. The code for each direction is as follows:

- LEVÁ:** turn left for 90 degrees, drive forward for 1 steps, turn left for 180 degrees, drive forward for 1 steps, turn right for 90 degrees.
- HORNÍ:** drive forward for 1 steps, turn left for 180 degrees, drive forward for 1 steps.
- DOLNÍ:** turn left for 180 degrees, drive forward for 1 steps, turn left for 180 degrees, drive forward for 1 steps, turn right for 180 degrees.
- PRAVÁ:** turn right for 90 degrees, drive forward for 1 steps, turn left for 180 degrees, drive forward for 1 steps, turn right for 90 degrees.

At the bottom left of the script is a yellow 'stop project' block.

Metodická doporučení:

- program je rozložen na jednotlivé části tak, aby žáci názorně viděli „společné“ části programů
- šedé bloky komentářů připravují a naznačují použití vlastních bloků
- připravené části programu lze nyní různě kombinovat a vytvářet řešení různých pohybů po vlajkách

Ukázka řešení s využitím vlastních bloků – řešení 1

The image displays a Scratch code editor with a custom block solution. The main script starts with a 'when started' block, followed by four custom blocks: 'LEVÁ', 'HORNÍ', 'DOLNÍ', and 'PRAVÁ'. Each custom block contains a 'turn' block with specific directions and degrees. The 'LEVÁ' block turns left 90 degrees, 'HORNÍ' turns left 180 degrees, 'DOLNÍ' turns right 180 degrees, and 'PRAVÁ' turns right 90 degrees. Below these are four 'TAM A ZPĚT' custom blocks, each containing a 'turn' block with specific directions and degrees: 'LEVÁ' turns right 90 degrees, 'HORNÍ' turns right 180 degrees, 'DOLNÍ' turns right 90 degrees, and 'PRAVÁ' turns right 90 degrees. The script ends with a 'stop project' block. A separate 'define' block for 'TAM A ZPĚT' is shown, containing three blocks: 'drive forward for 1 steps', 'turn left for 180 degrees', and 'drive forward for 1 steps'.

```
when started
  LEVÁ
  HORNÍ
  DOLNÍ
  PRAVÁ
  TAM A ZPĚT
  TAM A ZPĚT
  TAM A ZPĚT
  TAM A ZPĚT
  stop project

define TAM A ZPĚT
  drive forward for 1 steps
  turn left for 180 degrees
  drive forward for 1 steps
```

Ukázka řešení s využitím vlastních bloků – řešení 2

The image displays a Scratch script for a robot navigation task. The script begins with a 'when started' block, followed by a vertical stack of five custom blocks: 'LEVÁ', 'HORNÍ', 'DOLNÍ', 'PRAVÁ', and 'stop project'. The main script consists of five 'define' blocks, each followed by a sequence of actions:

- define LEVÁ:** turn left for 90 degrees, TAM A ZPĚT, turn right for 90 degrees.
- define HORNÍ:** TAM A ZPĚT.
- define DOLNÍ:** turn left for 180 degrees, TAM A ZPĚT, turn right for 180 degrees.
- define PRAVÁ:** turn right for 90 degrees, TAM A ZPĚT, turn right for 90 degrees.
- define TAM A ZPĚT:** drive forward for 1 steps, turn left for 180 degrees, drive forward for 1 steps.

Metodická doporučení:

- vytvořením vlastních vhodně pojmenovaných bloků dochází výraznému zpřehlednění kódu
- výrazně se zjednoduší zpřeházení pořadí průjezdu cílovými poli (vlaječkami) díky manipulaci s bloky

Ukázka řešení s využitím vlastních bloků – řešení 3

The image shows a Scratch script on a grid background. The script starts with a yellow 'when started' block, followed by a vertical stack of five pink 'define' blocks: 'LEVÁ', 'HORNÍ', 'DOLNÍ', 'PRAVÁ', and 'stop project'. The main script consists of several blue blocks: 'drive forward for 1 steps', 'turn left for 180 degrees', and 'drive forward for 1 steps'. There are also several custom blocks (pink) and their corresponding definitions (blue):

- 'define LEVÁ' is defined as 'VLEVO VBOOK', 'TAM A ZPĚT', and 'VPRAVO VBOOK'.
- 'define HORNÍ' is defined as 'TAM A ZPĚT'.
- 'define DOLNÍ' is defined as 'ČELEM VZAD', 'TAM A ZPĚT', and 'ČELEM VZAD'.
- 'define PRAVÁ' is defined as 'VPRAVO VBOOK', 'TAM A ZPĚT', and 'VLEVO VBOOK'.
- 'define VPRAVO VBOOK' is defined as 'turn left for 90 degrees'.
- 'define ČELEM VZAD' is defined as 'turn right for 180 degrees'.
- 'define VLEVO VBOOK' is defined as 'turn right for 90 degrees'.

Metodická doporučení:

- vytvořením vlastních bloků pro povely otáčení dojde k dalšímu zpřehlednění a čitelnosti kódu programu
- je nutné na závěr aktivity připomenout žákům vývoj programu od původní nepřehledné verze až do konečného řešení
- správný průběh vzniku programu by měl být ale obrácený – rozklad problému na subproblémy a jejich řešení

Vlastní bloky - 2

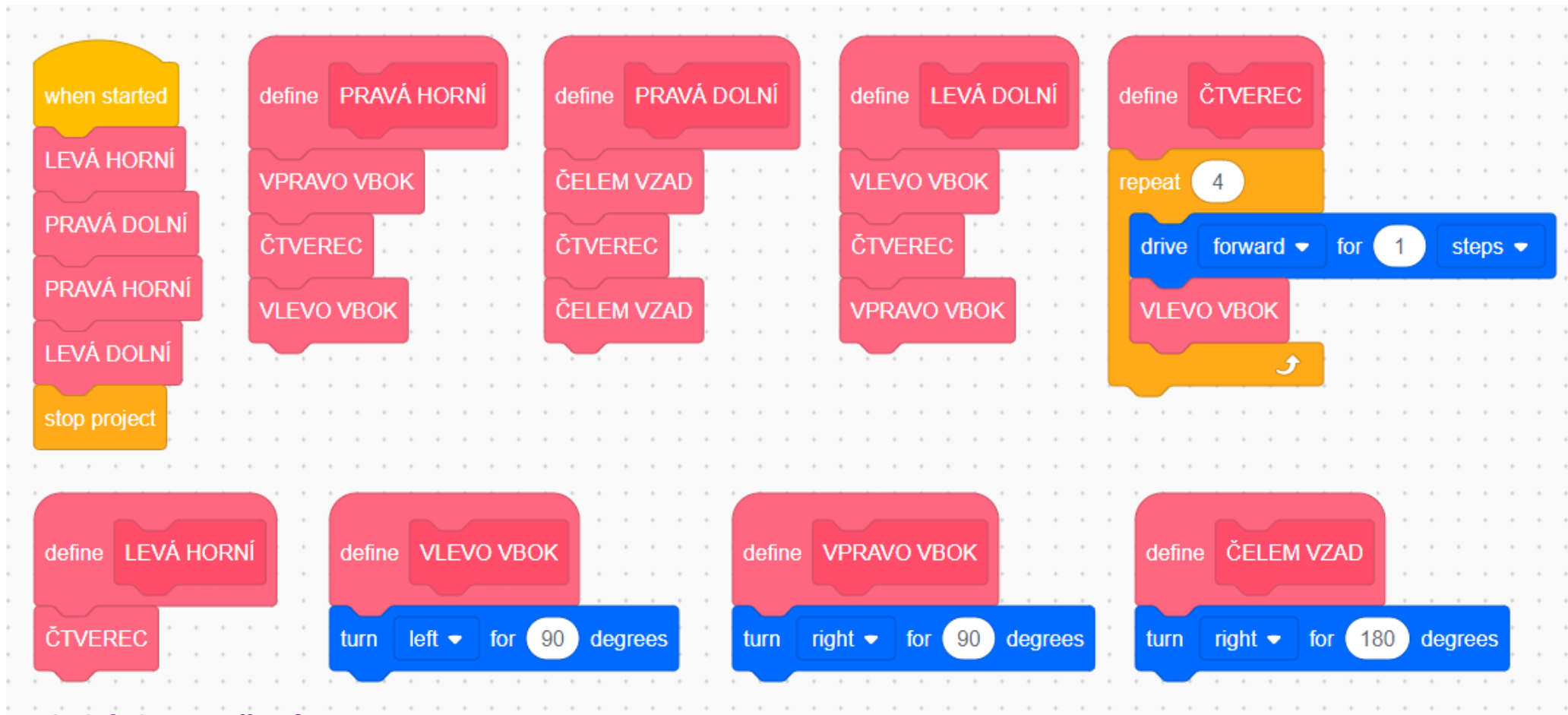


Vytvoř program, s jehož pomocí VEXík postupně dojde na cílová pole s vlajkou a zpět. Nejprve vše vyzkoušej na desce s maketou a program zapiš pomocí blokových příkazů do jednoho scénáře. Nakonec vyzkoušej VEXíka naprogramovat a všechny úkoly splň s ním.

Metodická doporučení:

- analogická úloha na využití vlastních bloků
- řešení lze provést metodou „tam a zpět“ (viz předchozí úloha) nebo metodou „malých čtverců“
- doporučuje se již využít od začátku rozkladu problému na dílčí problémy (opačný postup než v minulé úloze)

Ukázka řešení s využitím vlastních bloků a jednoduchého cyklu repeat

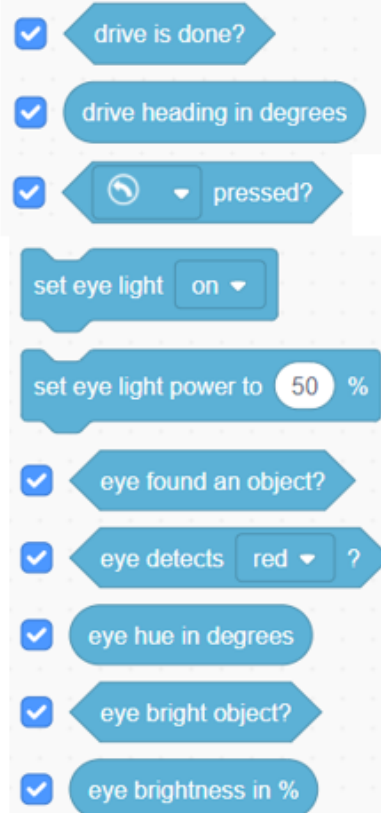


Metodická doporučení:

- řešení pomocí čtverců využívá použití jednoduchého cyklu **repeat** (4x opakuj strana + otočení o 90°)

7. ročník

Bloky pro využití senzorů



PŘIPRAVENÝ?

OTOČENÍ VEX ve stupních

TLAČÍTKO stisknuté?

ZAPNI / VYPNI OČI

NASTAV SVÍTIVOST na X%

NALEZEN OBJEKT?

DETEKOVÁNA BARVA?

ODSTÍN v %

DETEKOVÁN JASNÝ OBJEKT?

JAS v %

A screenshot of the VEXCode Monitor interface. The title bar is blue with the text 'Monitor' and a sun icon. Below the title bar is a table with the heading 'Sensors'. The table has two columns: the sensor name and its current value. The rows are: 'drive is done?' with value 'true', 'drive heading in degrees' with value '45', 'left button pressed?' with value 'false', 'eye found an object?' with value 'false', 'eye detects red?' with value 'false', 'eye hue in degrees' with value '40', 'eye bright object?' with value 'false', and 'eye brightness in %' with value '16'. Below the table is an orange bar with the heading 'Variables'. There is a blue 'AD' button on the right side of the interface.

Sensors	
drive is done?	true
drive heading in degrees	45
left button pressed?	false
eye found an object?	false
eye detects red?	false
eye hue in degrees	40
eye bright object?	false
eye brightness in %	16

Metodická doporučení:

- na základě zvládnutí učiva Scratch v 7. ročníku (podmíněný příkaz, cykly s podmínkou) lze začít využívat bloky pro práci s VEXíkovými senzory (optické čidlo, tlačítka) a ovládací Monitor ve webové aplikaci VEXCode

Cykly - ukázky

```
when started
repeat 10
  turn right for 90 degrees
```

1)

```
when started
forever
  wait until pressed?
  play sound honk
```

2)

```
when started
repeat until pressed?
  glow purple
glow off
stop project
```

3)

```
when started
while not pressed?
  glow purple
glow off
stop project
```

4)

Metodická doporučení:

- je vhodné žáky postupně seznámit se všemi druhy cyklů (ekvivalent bloku **while** žáci z prostředí Scratch neznají)
- u bloků s podmínkou je žádoucí vyzkoušet vykonání příkazu v cyklu na základě vyhodnocení podmínky
- v případě cyklu **while** žáky upozorněte na nutnost stisknutí tlačítka ještě před spuštěním programu
- žáci by měli vymýšlet alternativní použití bloků (s využitím svícení středového tlačítka nebo hraní zvuků)

Podmíněný příkaz - ukázky

Porovnejte činnost VEXíka v těchto 4 scénářích:

```
when started
forever
  if pressed? then
    glow green
  wait 1 seconds
  glow off
```

```
when started
forever
  if pressed? then
    glow green
  else
    glow off
```

```
when started
forever
  if pressed? then
    glow blue
  if pressed? then
    glow green
  if pressed? then
    glow purple
  glow off
```

```
when started
forever
  if pressed? then
    glow green
  else if pressed? then
    glow blue
  else if pressed? then
    glow purple
  else
    glow off
```

Metodická doporučení:

- v těchto přípravných úlohách je nutné klást důraz na pochopení opakovaného použití podmíněného příkazu a vnořeného strukturovaného podmíněného příkazu
- žáky opět necháme různě experimentovat se zvládnutím obou druhů podmíněných bloků

Využití optického senzoru - ukázky

Porovnejte činnost VEXíka v těchto 4 scénářích:

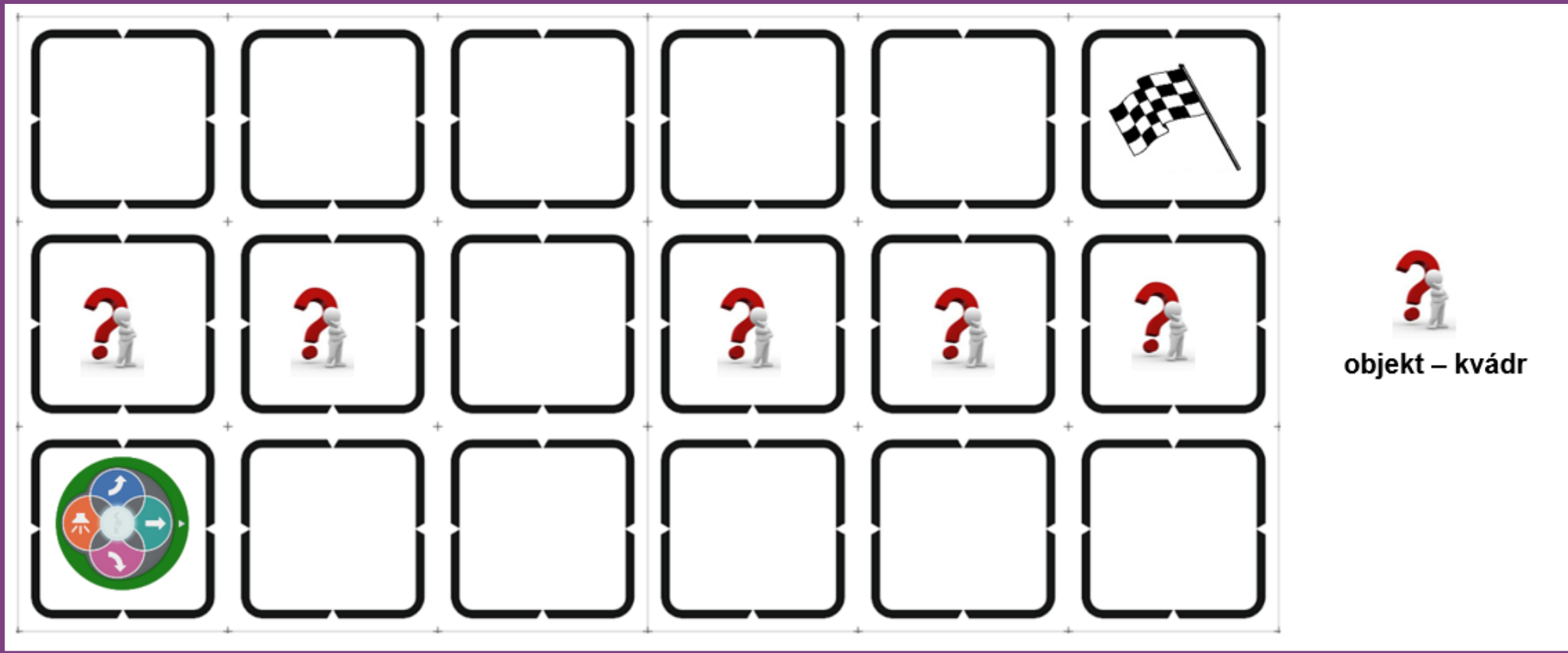
The image displays four distinct VEXcode block configurations for an optical sensor, each starting with a 'when started' block followed by a 'forever' loop:

- Scenario 1:** An 'if' block with the condition 'eye found an object?' is followed by a 'glow purple' block. Below it, an 'if' block with the condition 'not eye found an object?' is also followed by a 'glow purple' block.
- Scenario 2:** An 'if' block with the condition 'eye found an object?' is followed by a 'glow purple' block. An 'else' block is attached to the bottom of the 'if' block, containing a 'glow purple' block.
- Scenario 3:** Three sequential 'if' blocks are used. The first checks 'eye detects blue?' and triggers 'glow blue'. The second checks 'eye detects green?' and triggers 'glow green'. The third checks 'eye detects red?' and triggers 'glow purple'.
- Scenario 4:** An 'if' block checks 'eye detects blue?' (triggers 'glow blue'), followed by an 'else if' block for 'eye detects green?' (triggers 'glow green'), and another 'else if' block for 'eye detects red?' (triggers 'glow purple'). A final 'else' block contains a 'glow off' block.

Metodická doporučení:

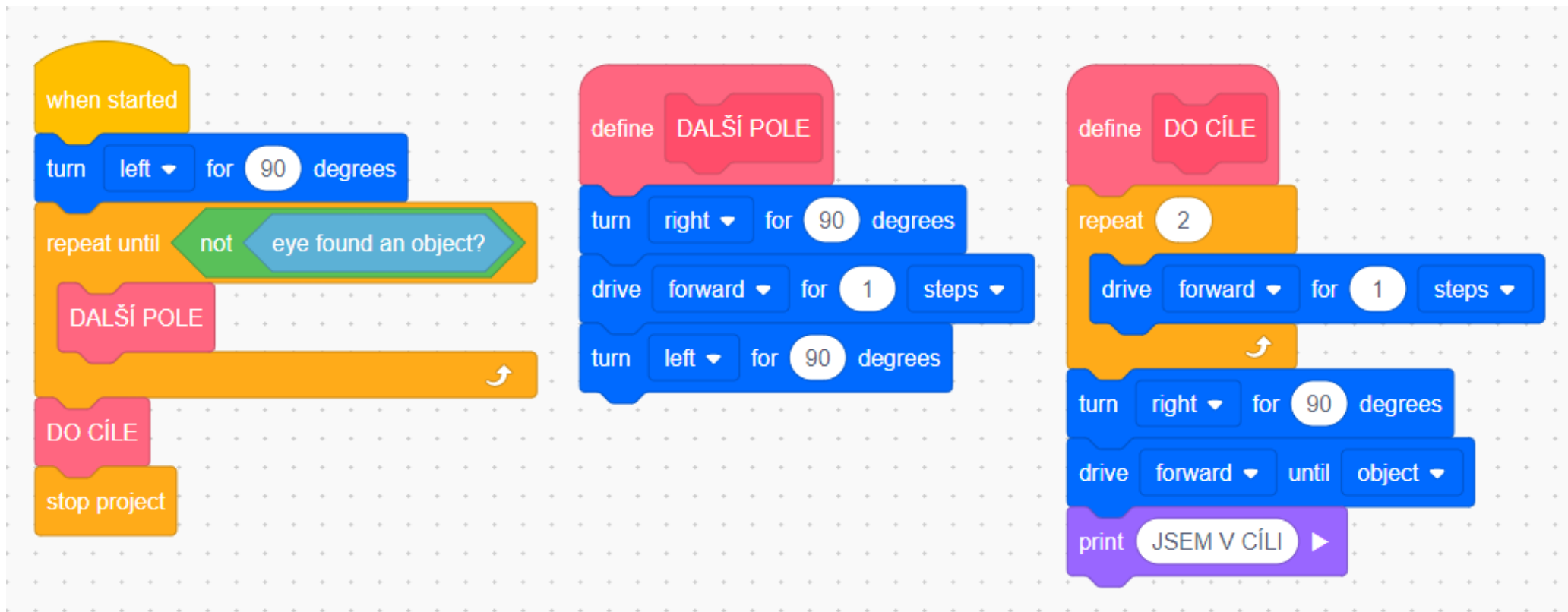
- pro experimentování s optickým čidlem VEXíka lze využít zpočátku barevných kroužků ze školní sady
- vzhledem k dalším pokročilejším úlohám je nutné vyrobit sadu barevných objektů (ideálně dřevěné hranoly 4×4×1 cm nebo na 3D tiskárně vytisknutý podobný předmět – stojánek 4×4 cm na podstavci)
- je důležité se žáky sledovat oznámení na Monitoru VEXCodu s výsledky činnosti optického senzoru

Hledání otvoru v řadě



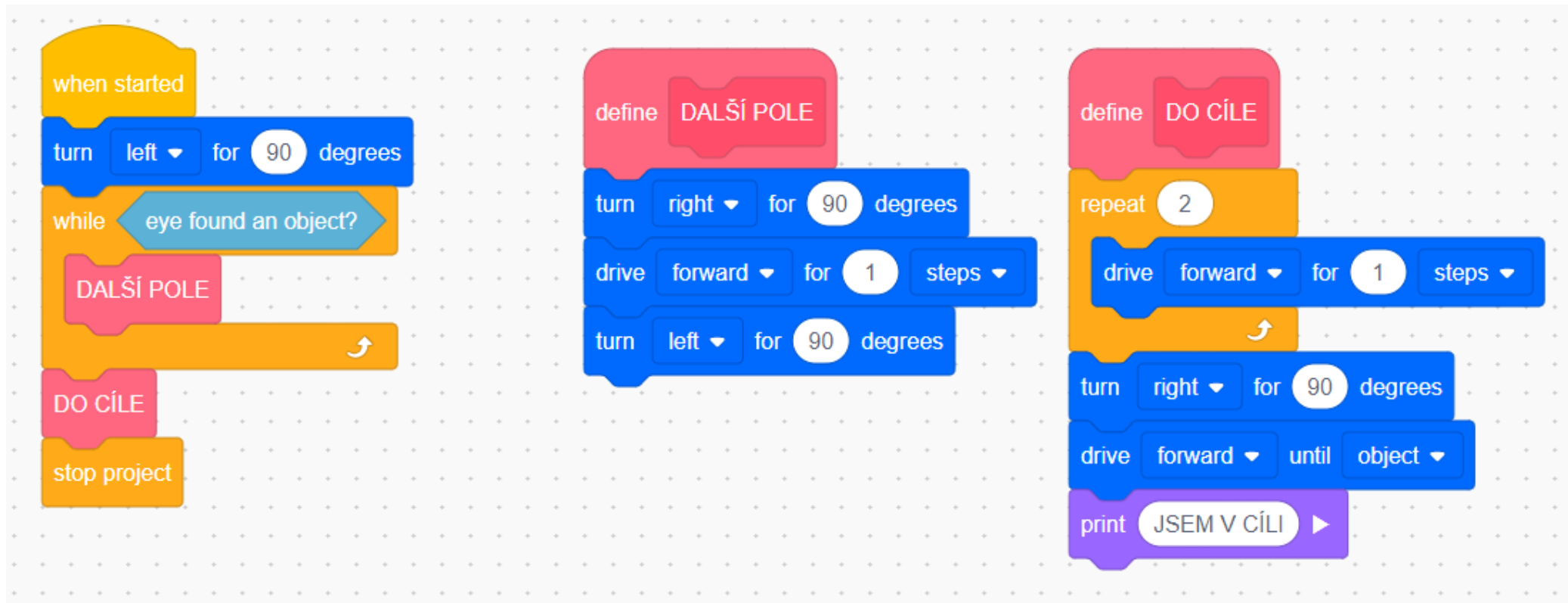
Metodická doporučení:

- VEXík má pomocí programu jet podél řady překážek a najít průjezd na druhou stranu řady, kde dojde do cíle u mantinelu



Metodická doporučení:

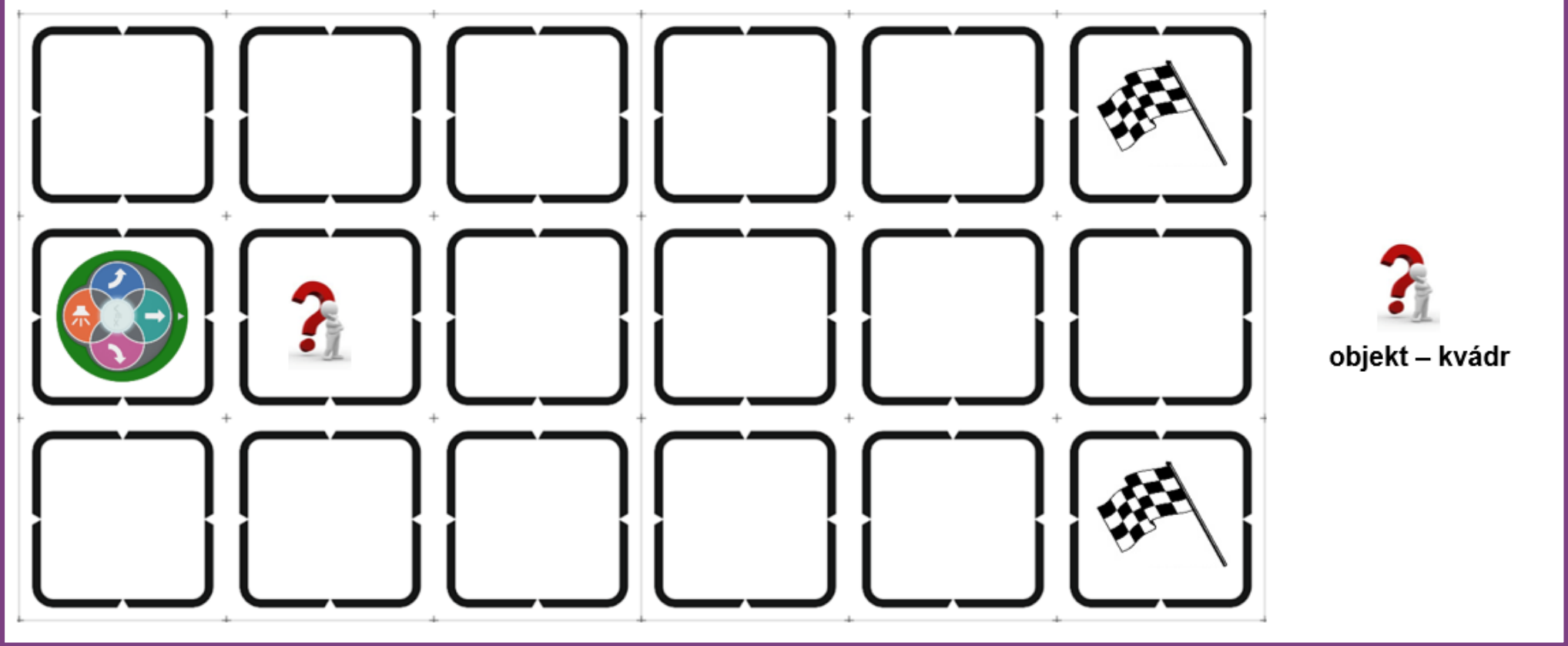
- řešení s využitím bloku **repeat until**



Metodická doporučení:

- řešení s využitím bloku **while**
- dokáží žáci pojmenovat rozdíl v použití obou podmíněných cyklů?

Semaforey 1



Metodická doporučení:

- sada příkladů SEMAFORY řeší pohyb VEXíka v závislosti na výskytu a barvě objektu na jeho cestě na cílové pole
- každý příklad přidává obtížnější podmínky pohybu
- je nutné nejprve provést rozbor pohybu (příprava vlastních bloků), pak jejich vytvoření a krokování programu

```

when started
K SEMAFORU
TEST SEMAFORU
print JSEM V CÍLI! and set cursor to next row
stop project

```

```

define K SEMAFORU
repeat until eye found an object?
drive forward for 1 steps

```

```

define TEST SEMAFORU
if eye detects red ? then
OBJEZD ZLEVA
else if eye detects blue ? then
OBJEZD ZPRAVA
else
stop driving

```

```

define OBJEZD ZLEVA
turn left for 90 degrees
drive forward for 1 steps
turn right for 90 degrees
drive forward until object

```

```

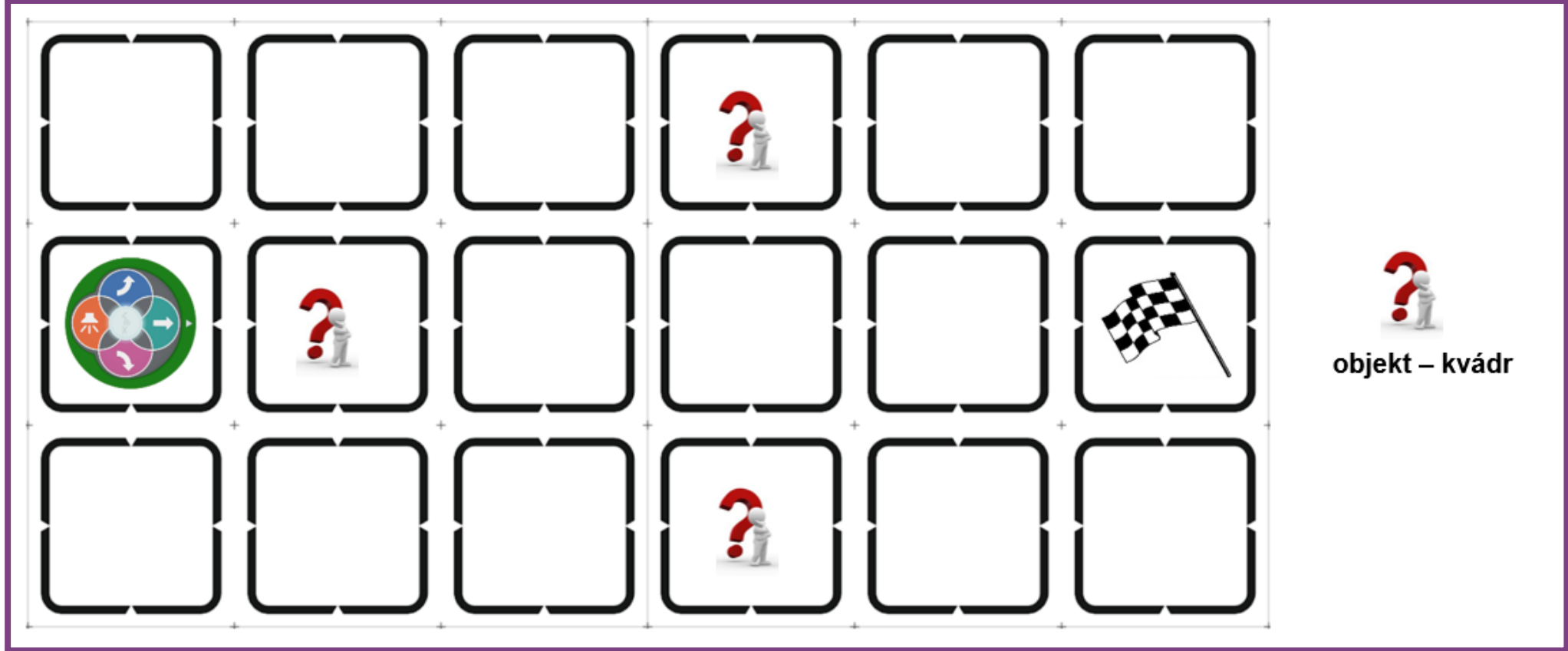
define OBJEZD ZPRAVA
turn right for 90 degrees
drive forward for 1 steps
turn left for 90 degrees
drive forward until object

```

Metodická doporučení:

- pokud bude žádoucí řešit umístění semaforu mimo přesné umístění v řadě polí, lze blok **repeat until** nahradit blokem **drive until** (řešení musí být vždy univerzální bez ohledu na pozici semaforu)

Semafony 2



Metodická doporučení:

- příklad řeší několik druhů pohybu – k semaforu, k překážce, k cíli
- řešení musí být natolik univerzální, aby VEXík reagoval správně na libovolnou pozici semaforu i překážky

```

when started
  K SEMAFORU
  TEST SEMAFORU
  DO CÍLE
  stop project

define TEST SEMAFORU
  if eye detects red? then
    OBJEZD ZLEVA
  else if eye detects blue? then
    OBJEZD ZPRAVA
  else
    stop driving

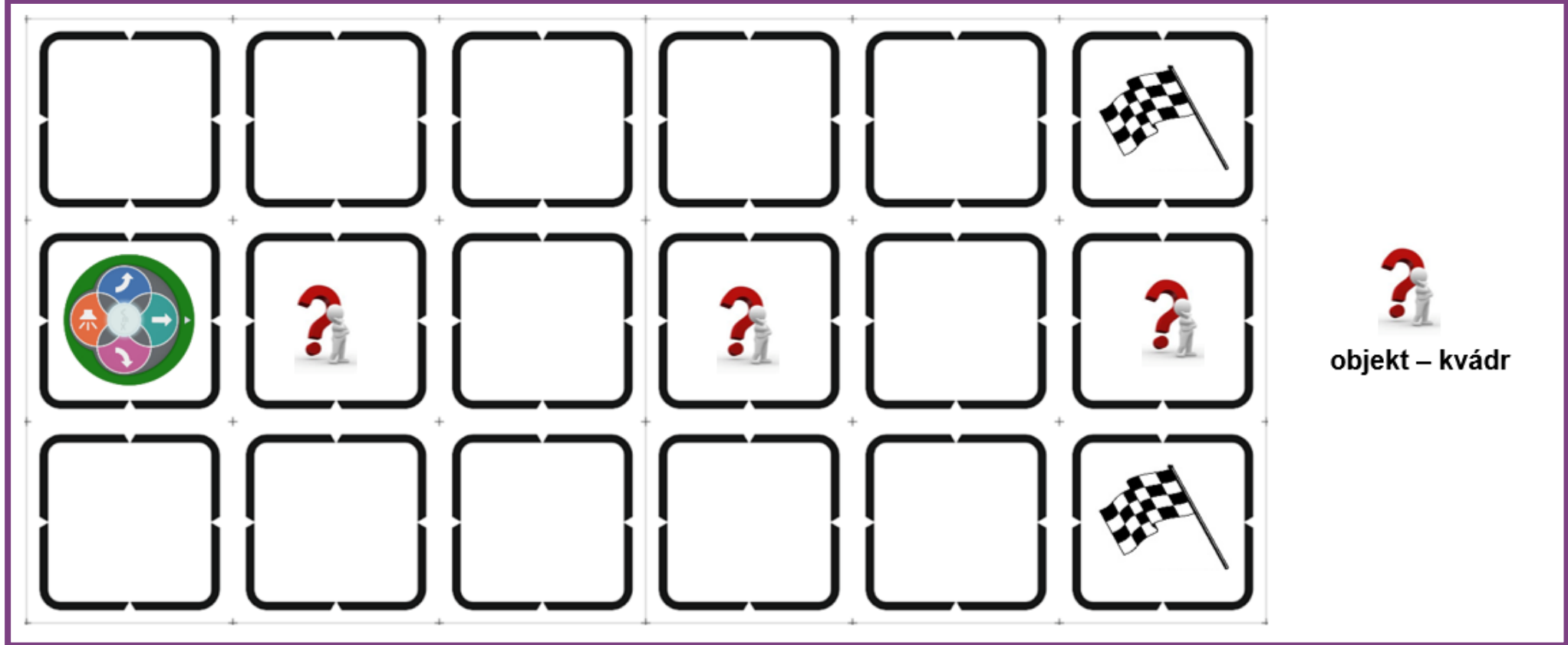
define OBJEZD ZLEVA
  turn left for 90 degrees
  drive forward for 1 steps
  turn right for 90 degrees
  drive forward until object
  turn right for 90 degrees
  drive forward for 1 steps
  turn left for 90 degrees

define OBJEZD ZPRAVA
  turn right for 90 degrees
  drive forward for 1 steps
  turn left for 90 degrees
  drive forward until object
  turn left for 90 degrees
  drive forward for 1 steps
  turn right for 90 degrees

define DO CÍLE
  drive forward until object
  print JSEM V CÍLI! and set cursor to next row

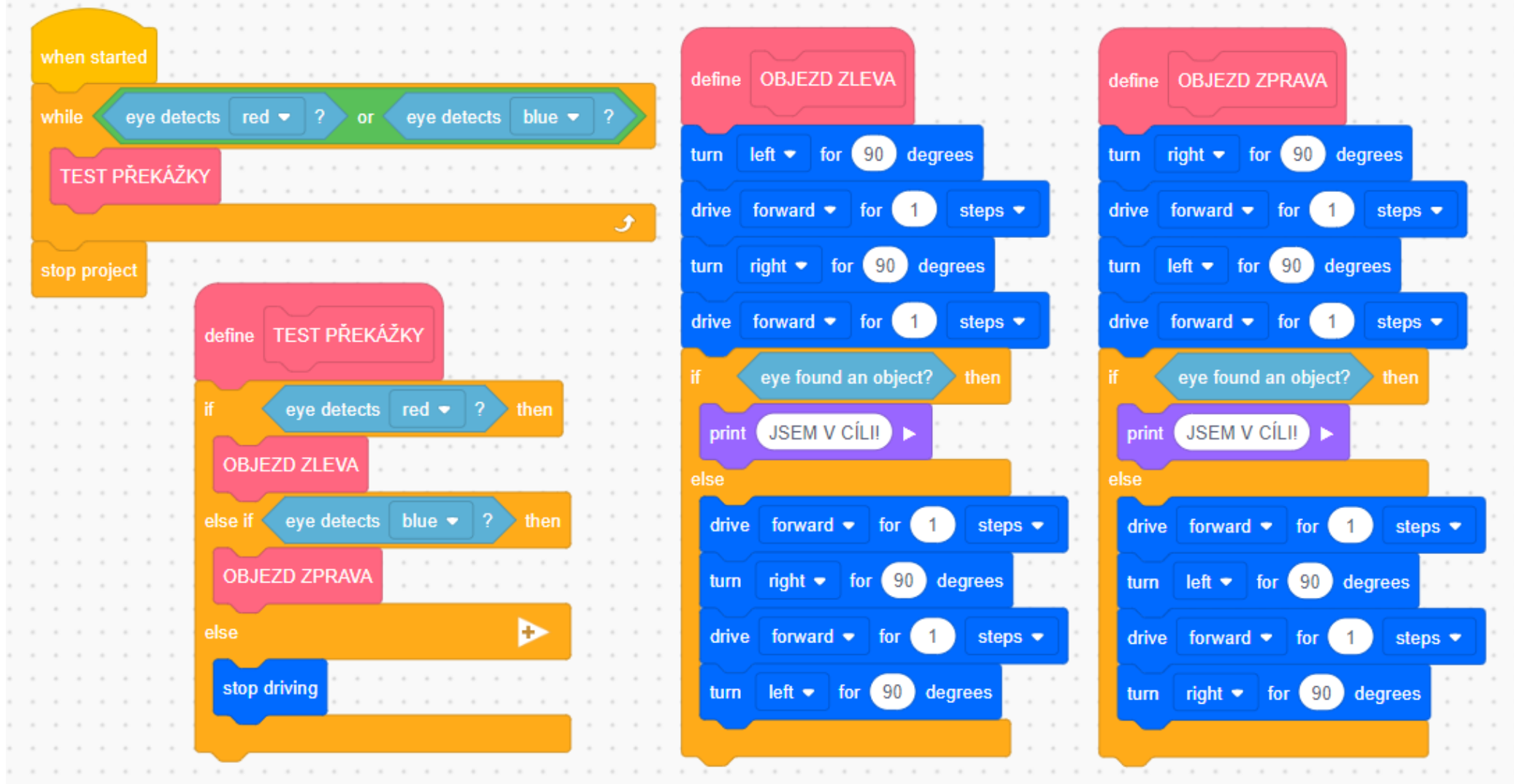
define K SEMAFORU
  drive forward until object
  
```

Semaforey 4



Metodická doporučení:










- příklad řeší již výskyt 3 semaforů, které bude objíždět zleva nebo zprava dle barvy na semaforu
- pro zastavení VEXíka na konci herního pole slouží 1 ks mantinelu, jehož přítomnost detekuje optický senzor




Metodická doporučení:

- největším problémem je zastavení VEXíka na cílovém poli s využitím výskytu mantinelu (tj. předmětu bez barvy)

Hledání parkovacího místa


objekt – kvádr

Metodická doporučení:

- VEXíkovým úkolem je projet mezi dvěma řadami parkovacích míst a do Monitoru zapsat přehled volných a obsazených parkovacích míst
- vzhledem k požadované univerzálnosti programu je pro ukončení VEXíkovy činnosti vyžadována reakce na mantinely


```

when started
  clear all rows
  repeat 2
    repeat until eye found an object?
      TEST PARKOVACÍHO MÍSTA
      drive forward for 1 steps
    turn right for 180 degrees
    print ◀ and set cursor to next row
  print PARKOVIŠTĚ PROZKOUMÁNO! ▶
  stop project

define TEST PARKOVACÍHO MÍSTA
  turn left for 90 degrees
  if not eye found an object? then
    print VOLNO ◀ and set cursor to next row
    ZAPARKOVÁNÍ
    play sound honk
    ODJEZD
  else
    turn right for 90 degrees
    print OBSAZENO ◀ and set cursor to next row

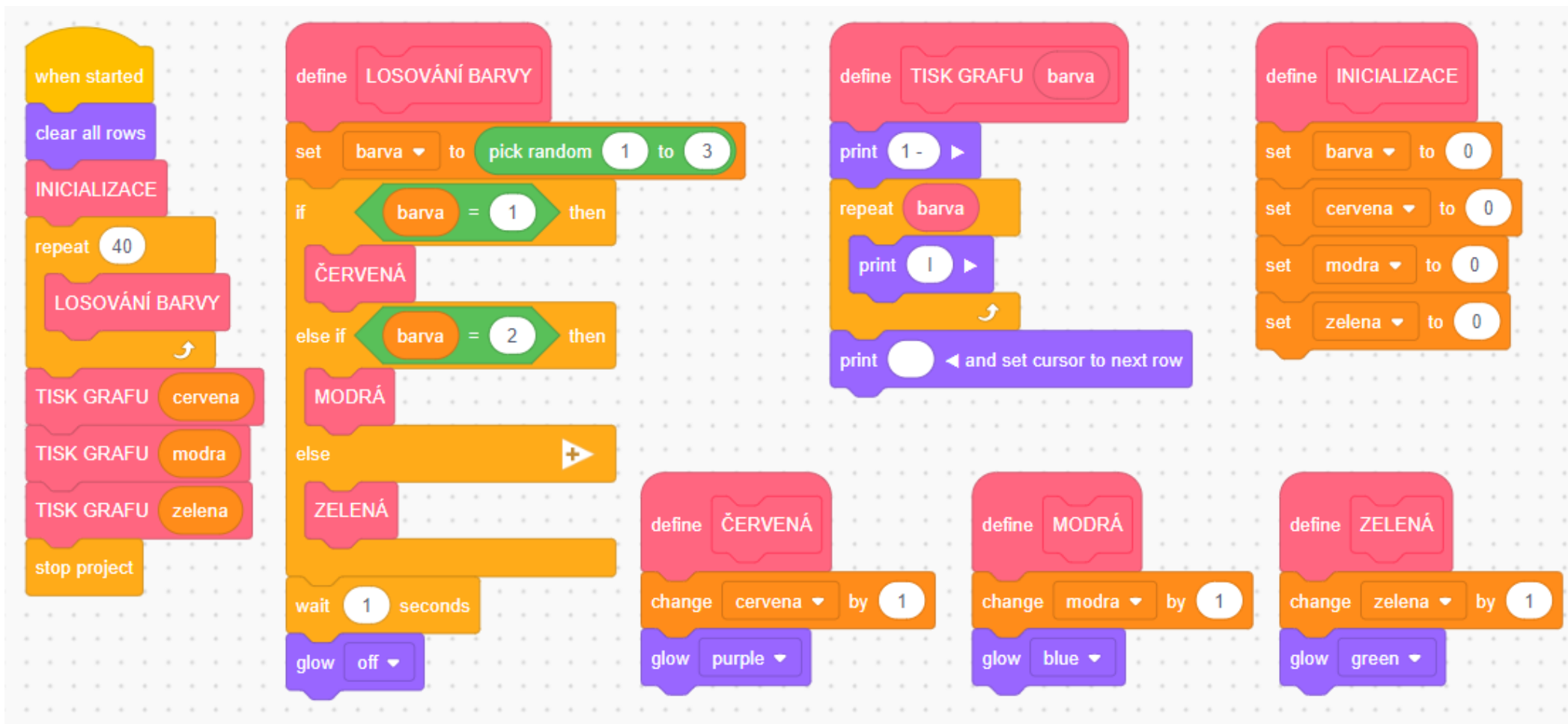
define ZAPARKOVÁNÍ
  drive forward for 1 steps
  turn right for 180 degrees

define ODJEZD
  drive forward for 1 steps
  turn left for 90 degrees

```

8. ročník

Opětovným propojením oblastí **Algoritmizace a Data, informace** může být zařazení úlohy na vytvoření programu, kdy VEXík bude generovat sérii náhodných čísel od 1 do 3, která se pomocí podmíněného příkazu budou „překládat“ do barev, kterými má VEXík svítit. Úlohu budeme řešit tentokrát s využitím proměnných, se kterými se již žáci seznámili minimálně v matematice, případně i v kurzu Scratch. Žáci pak tuto skupinu náhodných čísel budou vizualizovat pomocí pruhového pseudografu tiskem znaků v Monitoru. Na závěr aktivity můžeme opět provést společnou vizualizaci třídního souboru dat pomocí online nástroje <https://www.chartgo.com/index.jsp> a porovnávat výsledky náhodného generování (pravděpodobnost a statistika):



Monitor

Sensors	

Variables	
barva	3
cervena	16
modra	15
zelená	9

ADD LISTS

```

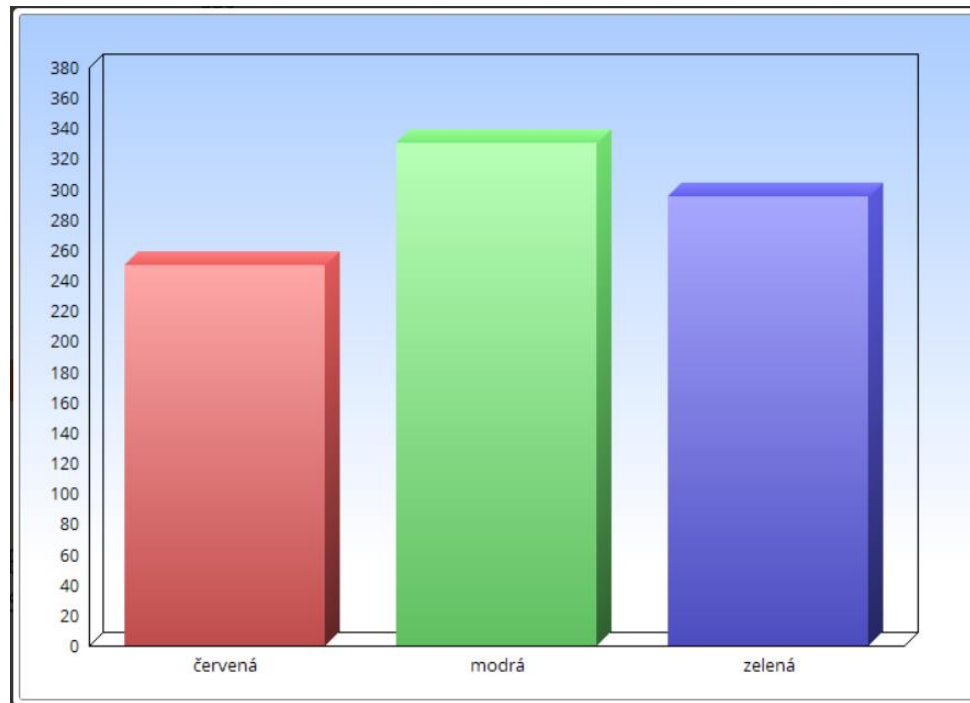
1 - 1111111111111111
1 - 1111111111111111
1 - 111111111

```

CLEAR SAVE COPY TO CLIPBOARD

Při řešení úlohy stále dbáme na uvědomělé využívání vlastních bloků, zjednodušující orientaci v programovém kódu. Je nutné také žákům vysvětlit význam INICIALIZACE proměnných na začátku programu. Celý průběh programu, generování čísel a plnění obsahu proměnných lze velmi dobře sledovat právě v Monitoru, kde na závěr pomocí cyklů a používáním vlastních bloků s parametrem vytvoříme také pruhový pseudograf. Pro potlačení vlivu pravděpodobnosti je vhodné nastavit větší číslo cyklování v hlavním bloku programu. Žáky vyzveme k formulování domněnek ohledně pravděpodobnosti generování jednotlivých čísel.

Podobnou úlohu lze vytvořit v rámci procvičování také pro simulaci hodu kostkou, změnami jsou pak zvýšení počtu proměnných, vlastních bloků a řádků (sloupců) získaného grafu.

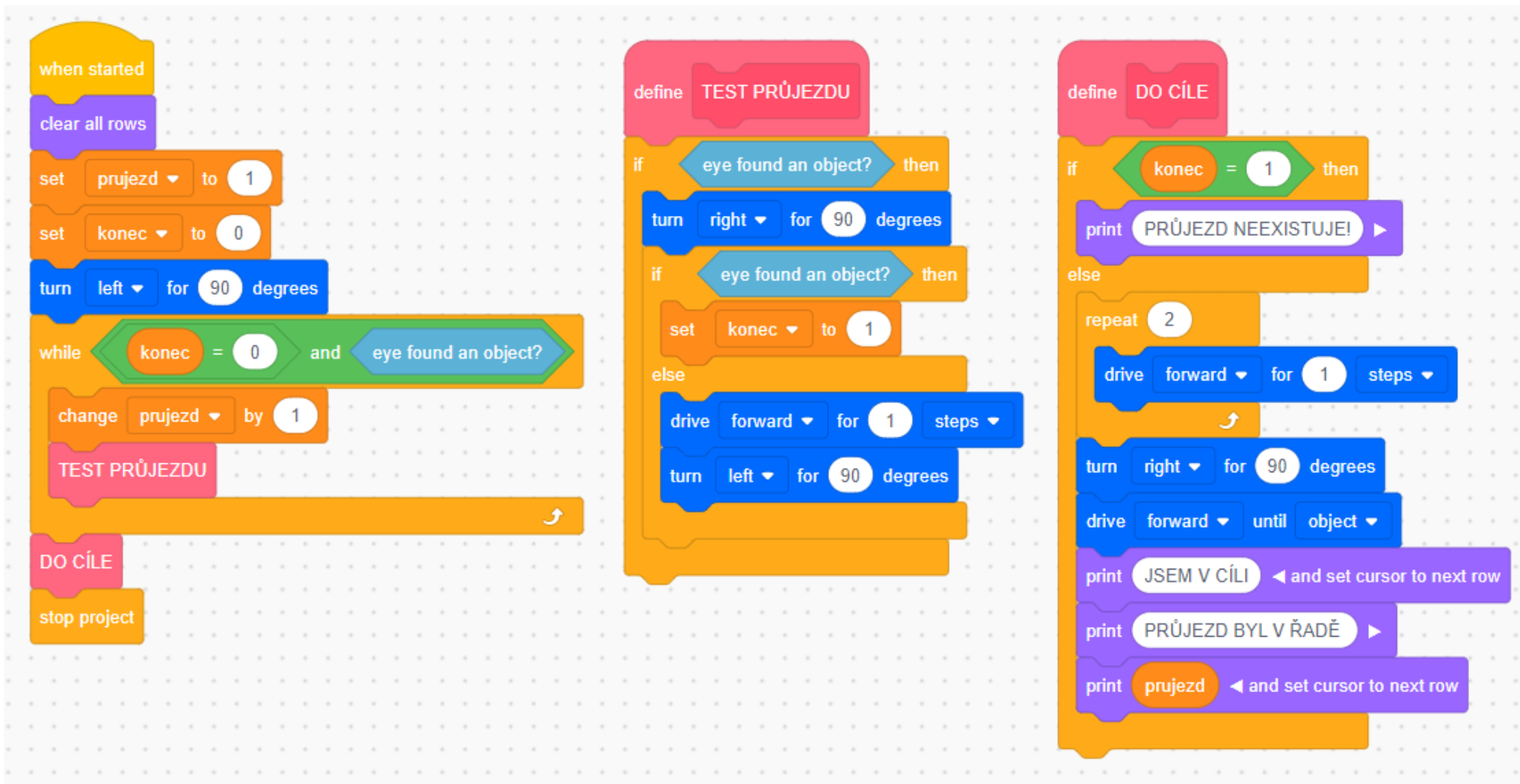


Hledání otvoru v řadě

objekt – kvádr

Metodická doporučení:

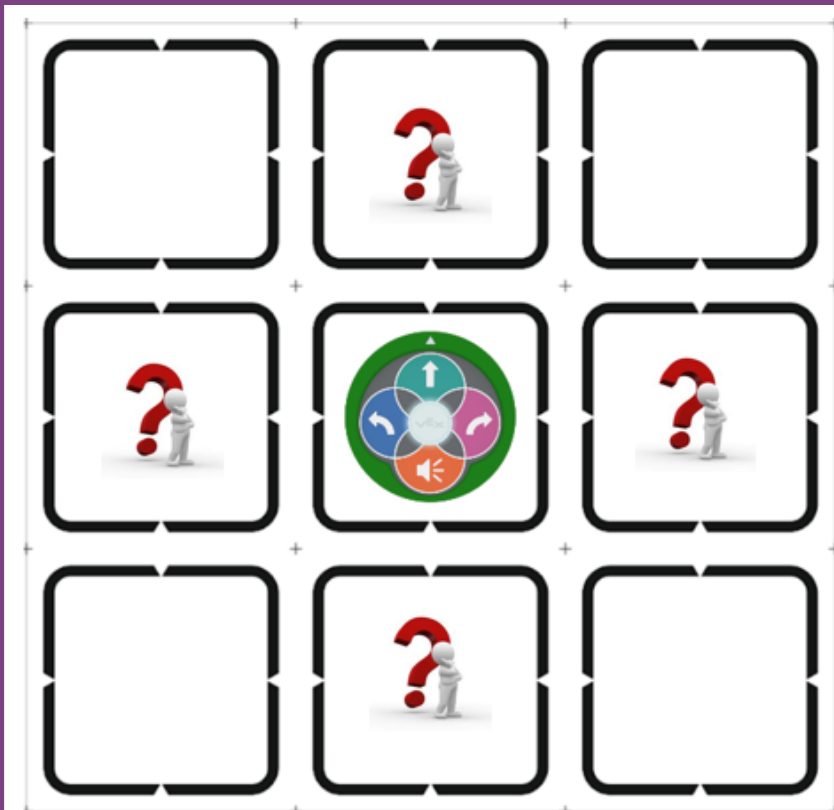
- VEXík má jet podél řady překážek a najít průjezd (existuje-li) na druhou stranu řady, kde dojde do cíle u mantinelu
- na závěr VEXík vypíše číslo řady, ve kterém našel průjezd (úloha rozšiřuje stejnou úlohu ze 7. ročníku o použití proměnné)



Metodická doporučení:

- řešení používá podmíněný cyklus **while** – je vhodné vyzkoušet také řešení pomocí cyklu **repeat** 😊

Počítání barev - 1



Vytvoř program, s jehož pomocí VEXík postupně spočítá počet barevných kostiček ve svém okolí. Nejprve vše vyzkoušej na desce s maketou a program zapiš pomocí blokových příkazů do jednoho scénáře. Nakonec vyzkoušej VEXíka naprogramovat a všechny úkoly splň s ním.

Zobrazení výsledků bude probíhat v informačním okně systému code123.vex.com.

Metodická doporučení:

- žáci musí nadeklarovat vhodné proměnné, používat podmíněný blok a cyklus
- při vhodné velikosti barevných kvádrů jich lze postavit kolem VEXíka osm – jak se změní kód programu?

```
when started
  clear all rows
  set pocet_cervenych to 0
  set pocet_modrych to 0
  set pocet_zelenych to 0
  set pocet_bezbarvych to 0
  repeat 4
    test_kostky
    turn right for 90 degrees
  vystup
  stop project

define test_kostky
  test_cervene
  test_modre
  test_zelene

define test_modre
  if eye detects blue ? then
    change pocet_modrych by 1

define test_zelene
  if eye detects green ? then
    change pocet_zelenych by 1

define test_cervene
  if eye detects red ? then
    change pocet_cervenych by 1

define vystup
  print pocet_cervenych -
  print pocet_cervenych and set cursor to next row
  print pocet_zelenych -
  print pocet_zelenych and set cursor to next row
  print pocet_modrych -
  print pocet_modrych and set cursor to next row
  print pocet_bezbarvych -
  print pocet_bezbarvych and set cursor to next row
```


Počítání barev - 2



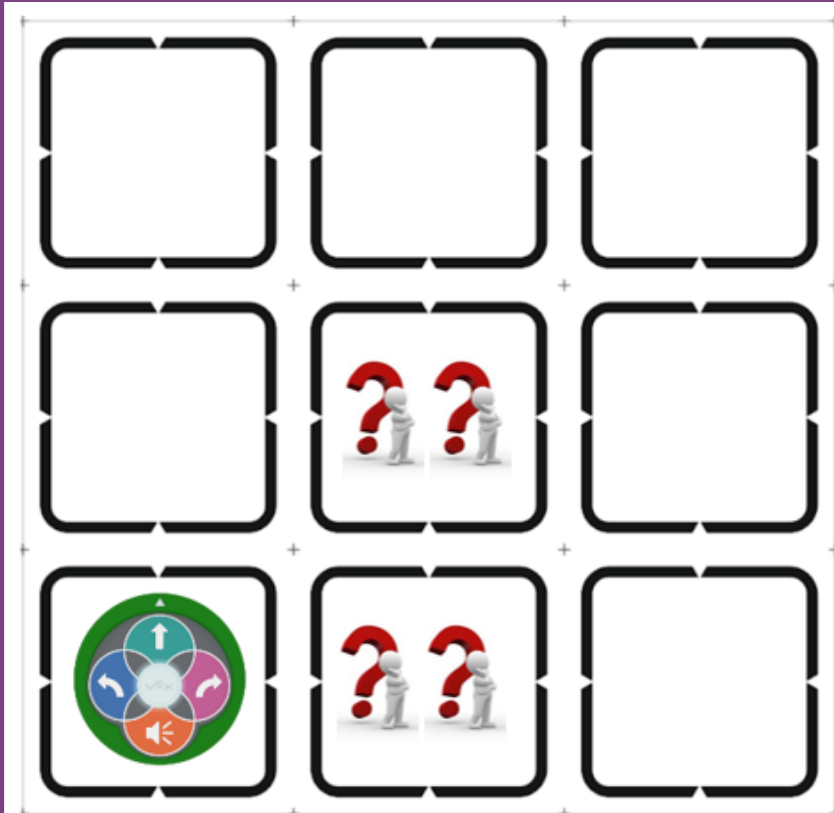
Vytvoř program, s jehož pomocí VEXík postupně spočítá počet barevných kostiček ve svém okolí. Nejprve vše vyzkoušej na desce s maketou a program zapiš pomocí blokových příkazů do jednoho scénáře. Nakonec vyzkoušej VEXíka naprogramovat a všechny úkoly splň s ním.

Zobrazení výsledků bude probíhat v informačním okně systému code123.vex.com.

Metodická doporučení:

- zadání úlohy je alternativním rozmístěním barevných kvádrů, používá řadu stejných postupů jako předchozí úlohy
- je vhodné se žáky vyřešit oba možné průjezdy – tam a zpět nebo jen tam s otáčením na obě strany
- univerzálnost řešení lze dokázat prodloužením herního pole o další desku a barevné kvádríky

Počítání barev - 3



Vytvoř program, s jehož pomocí VEXík postupně spočítá počet barevných kostiček ve svém okolí. Nejprve vše vyzkoušej na desce s maketou a program zapiš pomocí blokových příkazů do jednoho scénáře. Nakonec vyzkoušej VEXíka naprogramovat a všechny úkoly splň s ním.








Zobrazení výsledků bude probíhat formou blikání tlačítka VEX na VEXíkovi po stisku tlačítka KLAKSON.


Upravte řešení na libovolný počet vyhodnocovaných řad.

Metodická doporučení:

- zadání úlohy je alternativním rozmístěním barevných kvádrů, používá řadu stejných postupů jako předchozí úlohy
- opět je zapotřebí vyřešit způsob průjezdu VEXíka hracím polem, které lze opět nastavit další deskou a kvádříky

Počítání barev - 4


objekt – kvádr

Metodická doporučení:

- další varianta na počítání objektů a barev

```

when started
  uvod
  repeat 2
    while otooka = 0
      průzkum
      test_otocky
      if otooka = 0 then
        drive forward for 2 steps
      set otooka to 0
      turn right for 180 degrees
  konec
  stop project

define uvod
  set objekty to 0
  set cervene to 0
  set zelene to 0
  set otooka to 0
  drive forward for 1 steps

define průzkum
  turn left for 90 degrees
  drive forward for 1 steps
  turn left for 90 degrees

test_barev
  turn right for 180 degrees
test_barev
  turn left for 90 degrees
  drive forward for -1 steps
  turn right for 90 degrees

define test_otocky
  if eye found an object? then
    change otooka by 1

define konec
  print počet cervenych:
  print cervene and set cursor to next row
  print počet zelenych:
  print zelene and set cursor to next row
  print počet cervenych:
  print cervene and set cursor to next row
  print počet modrych:
  print cervene and set cursor to next row
  
```

9. ročník

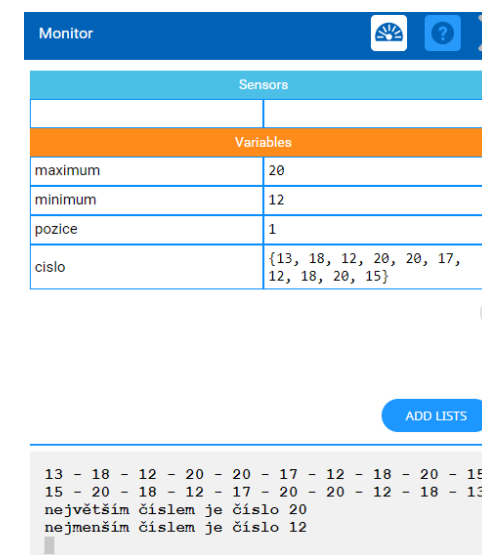
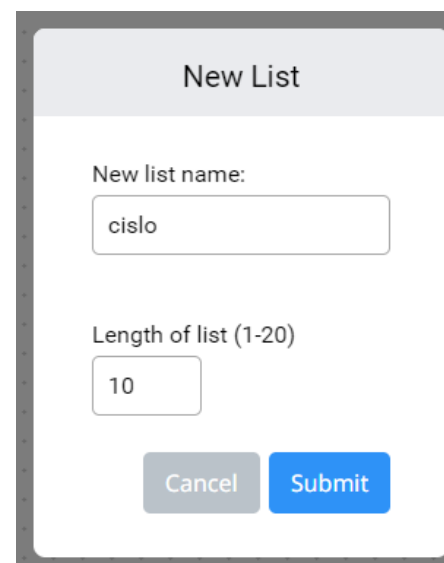
V rámci výuky algoritmizace a programování (Scratch) se žáci v rámci jednoho projektu setkají s využitím seznamu. Datová struktura **seznam (pole)** umožňuje ukládání většího množství logicky souvisejících dat (řady čísel např. hmotnost a výška žáků, známky z předmětu atd.) bez nutnosti tvorby řady proměnných. Seznamy mohou být vícerozměrné, pro potřebu řešení úloh s VEXíkem budeme využívat jednorozměrných a dvourozměrných seznamů (polí), které si můžeme představit jako jednořádkovou nebo víceřádkovou tabulku s větším počtem sloupců. Všechny položky seznamů mají stejný název, doplněný o index (pozici, pořadí v seznamu).

Na úvod se žáky probereme jednoduché využití seznamů (polí) bez návaznosti na VEXíkové pohyby nebo senzory, následně formou opakování a zdokonalování programových kódů ze 7. a 8. ročníku vyřešíme řadu úloh s pohyby a využitím senzorů (výuka ve spirále).

Nejprve musí žáci pochopit deklaraci seznamu (pole), pak vkládání čísel do jednotlivých položek a nakonec používání obsahu položek (např. pro tisk nebo výpočty). Vzhledem k tomu, že blokový jazyk VEXCode nezná klasický příkaz **for to** s implementovaným indexem, je nutné tuto záležitost zvládnout pomocnou proměnnou a postupnou změnou její velikosti. Tyto přípravné úlohy lze opět propojit s oblastí Data, informace.

V první přípravné úloze VEXík vygeneruje 10 náhodných čísel, které postupně uloží do seznamu, dále najde největší a nejmenší číslo. Nakonec vytiskne do Monitoru celý seznam vygenerovaných čísel od prvního i od posledního prvku, největší a nejmenší číslo.

Úlohy lze modifikovat požadavky na počet a pořadí tisknutých čísel – např. 1. polovina seznamu na jeden řádek, 2. polovina na druhý, druhá polovina seznamu opačně, 1. řádek čísla na sudých pozicích, druhý řádek čísla na lichých pozicích atd.



Sensors	

Variables	
maximum	20
minimum	12
pozice	1
cislo	{13, 18, 12, 20, 20, 17, 12, 18, 20, 15}

ADD LISTS

```
13 - 18 - 12 - 20 - 20 - 17 - 12 - 18 - 20 - 15
15 - 20 - 18 - 12 - 17 - 20 - 20 - 12 - 18 - 13
největším číslem je číslo 20
nejmenším číslem je číslo 12
```

when started
clear all rows
INICIALIZACE
NAPLNĚNÍ SEZNAMU
MAX a MIN
TISK tam
TISK zpět
TISK maxmin
stop project

define INICIALIZACE
set cislo to 0 0 0 0 0 0 0 0 0 0 0
set maximum to 0
set minimum to 21
set pozice to 0

define NAPLNĚNÍ SEZNAMU
repeat 10
change pozice by 1
replace item pozice of cislo to pick random 11 to 20

define MAX a MIN
set pozice to 0
repeat 10
change pozice by 1
if item pozice of cislo > maximum then
set maximum to item pozice of cislo
if item pozice of cislo < minimum then
set minimum to item pozice of cislo

define TISK tam
set pozice to 0
repeat 10
change pozice by 1
print item pozice of cislo
print -
print 0 and set cursor to next row

define TISK zpět
set pozice to 11
repeat 10
change pozice by -1
print item pozice of cislo
print -
print 0 and set cursor to next row

define TISK maxmin
print největším číslem je číslo
print maximum and set cursor to next row
print nejmenším číslem je číslo
print minimum and set cursor to next row

New 2D List

New 2D list name:

Height (rows) of 2D list (1-20)

Width (columns) of 2D list (1-20)

Monitor

Sensors	
Variables	
radek	3
pozice	0
cislo	{15, 18, 10, 10, 13, 17, 16, 15, 16, 15} {29, 26, 22, 22, 27, 28, 25, 29, 28, 27} {38, 35, 33, 30, 39, 33, 37, 31, 38, 35}

```

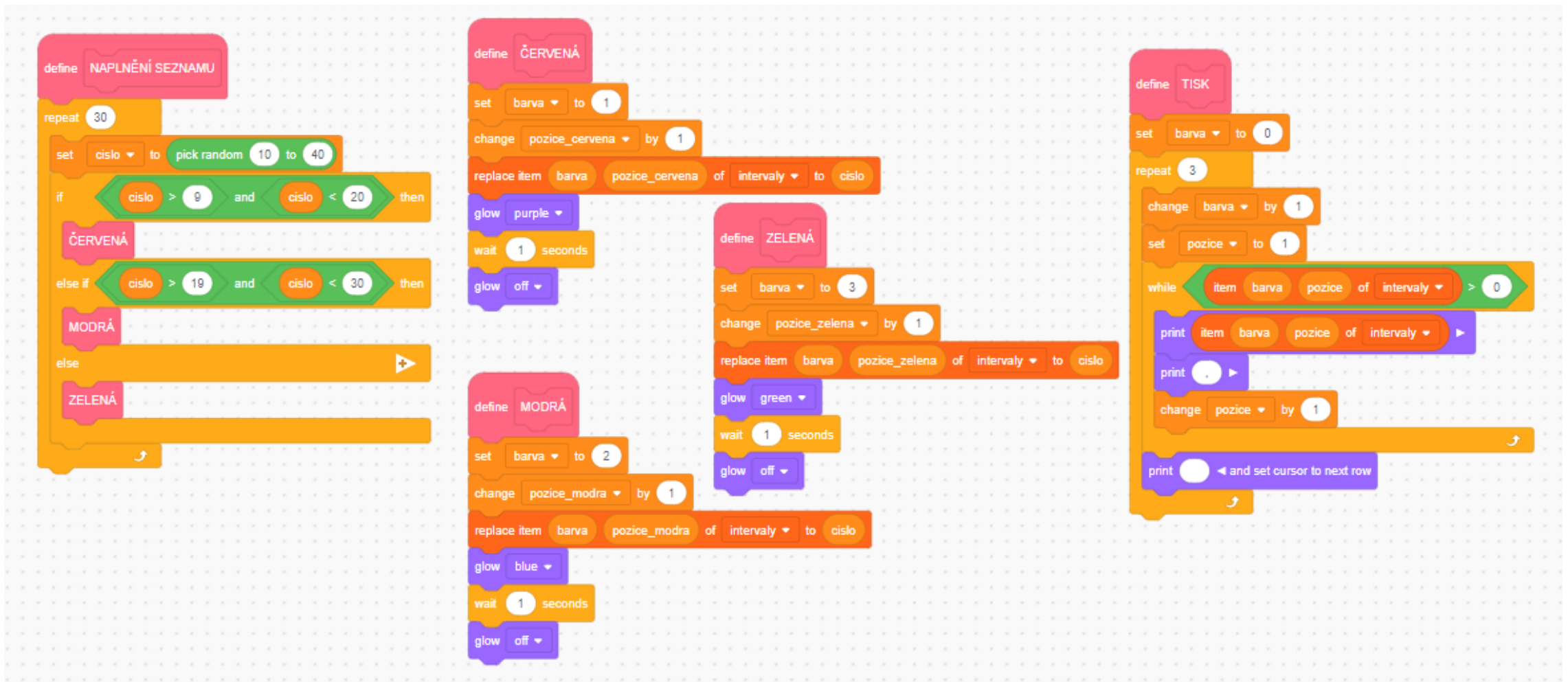
15 - 18 - 10 - 10 - 13 - 17 - 16 - 15 - 16 - 15
29 - 26 - 22 - 22 - 27 - 28 - 25 - 29 - 28 - 27
38 - 35 - 33 - 30 - 39 - 33 - 37 - 31 - 38 - 35
          
```


Ve druhé přípravné úloze VEXík vygeneruje 30 náhodných čísel, která postupně uloží do dvourozměrného seznamu (pole) podle jednoho ze tří intervalů, přičemž po uložení čísla vždy na 1 sekundu zasvíťí barvou intervalu. Nakonec vytiskne do Monitoru celý dvourozměrný seznam (pole) – tabulku s rozdělenými vygenerovanými čísly. Při řešení úlohy je zapotřebí si uvědomit, který z programovacích bloků bude v jednotlivých částech programu využit – v rámci generování 30 čísel (**repeat**) bude prováděno rozdělení podle velikosti čísla do intervalů (**if**), v rámci tisku se bude opakovat (**repeat**) tisk čísel v řádku dokud (**while**) budou položky větší než 0. Náročná je také práce s proměnnými, které fungují jako ukazatelé pozice zápisu nebo čtení dat z dvourozměrné tabulky.

Je vhodné začít ladění programu na menším číselném rozsahu (9 čísel – 3 intervaly), aby žáci mohli krokovat program. Toto krokování programu by jim pak mělo pomoci při pochopení činnosti programu.

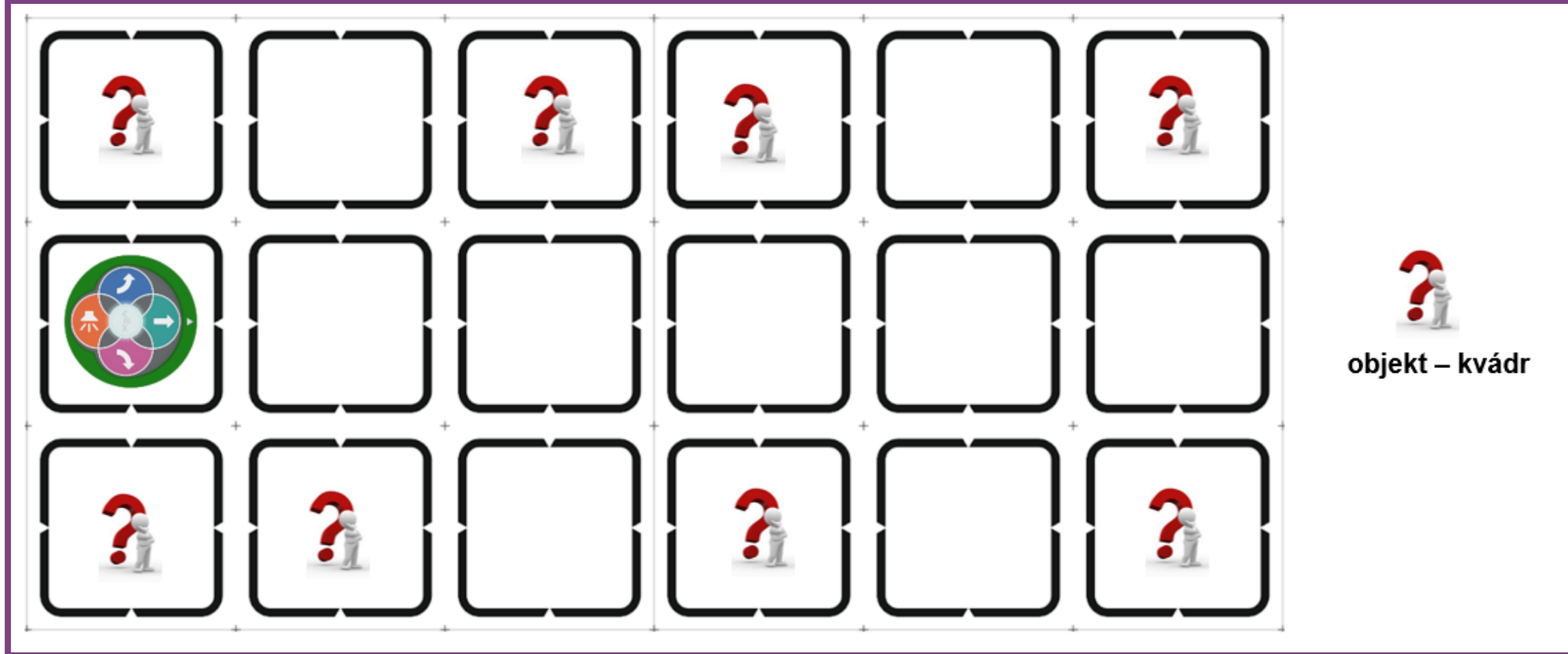
Monitor	
pozice_cervena	10
pozice_modra	12
pozice_zelena	8
cislo	33
intervaly	{14, 13, 17, 17, 17, 19, 16, 19, 10, 15, 0, 0, 0, 0, 0, 0, 0, 0}

14, 13, 17, 17, 17, 19, 16, 19, 10, 15,
20, 20, 20, 20, 23, 29, 20, 25, 28, 25, 23, 28,
37, 39, 40, 38, 32, 31, 32, 33,



Následující úlohy navazují na úlohy z předchozích ročníků a jsou zpravidla jejich rozšířením a doplněním o práci se seznamem (polem, tabulkou). Žáci tak ve spirálové výuce opakují, upevňují a rozšiřují své dovednosti. Úlohy v 9. ročníku jsou již poměrně náročné a zpravidla by se měly řešit formou společné práce a diskuze.

Hledání parkovacího místa



Metodická doporučení:

- kromě zjištění volných a obsazených parkovacích míst VEXík tak do Monitoru zapíše přesný rozpis míst na parkovišti (jednoduchý seznam s rozděleným tiskem do dvou řádků)

```

when started
  clear all rows
  set pozice to 1
  set parkoviste to 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  repeat 2
    repeat 6
      TEST PARKOVACÍHO MÍSTA
      drive forward for 1 steps
      change pozice by 1
    repeat 6
      print item pozice of parkoviste
      print 
      change pozice by 1
    print 
    and set cursor to next row
  set pozice to 12
  repeat 6
    print item pozice of parkoviste
    print 
    change pozice by -1
  print 
  and set cursor to next row

  TISK
  print PARKOVIŠTĚ PROZKOUMÁNO! and set cursor to next row
  stop project

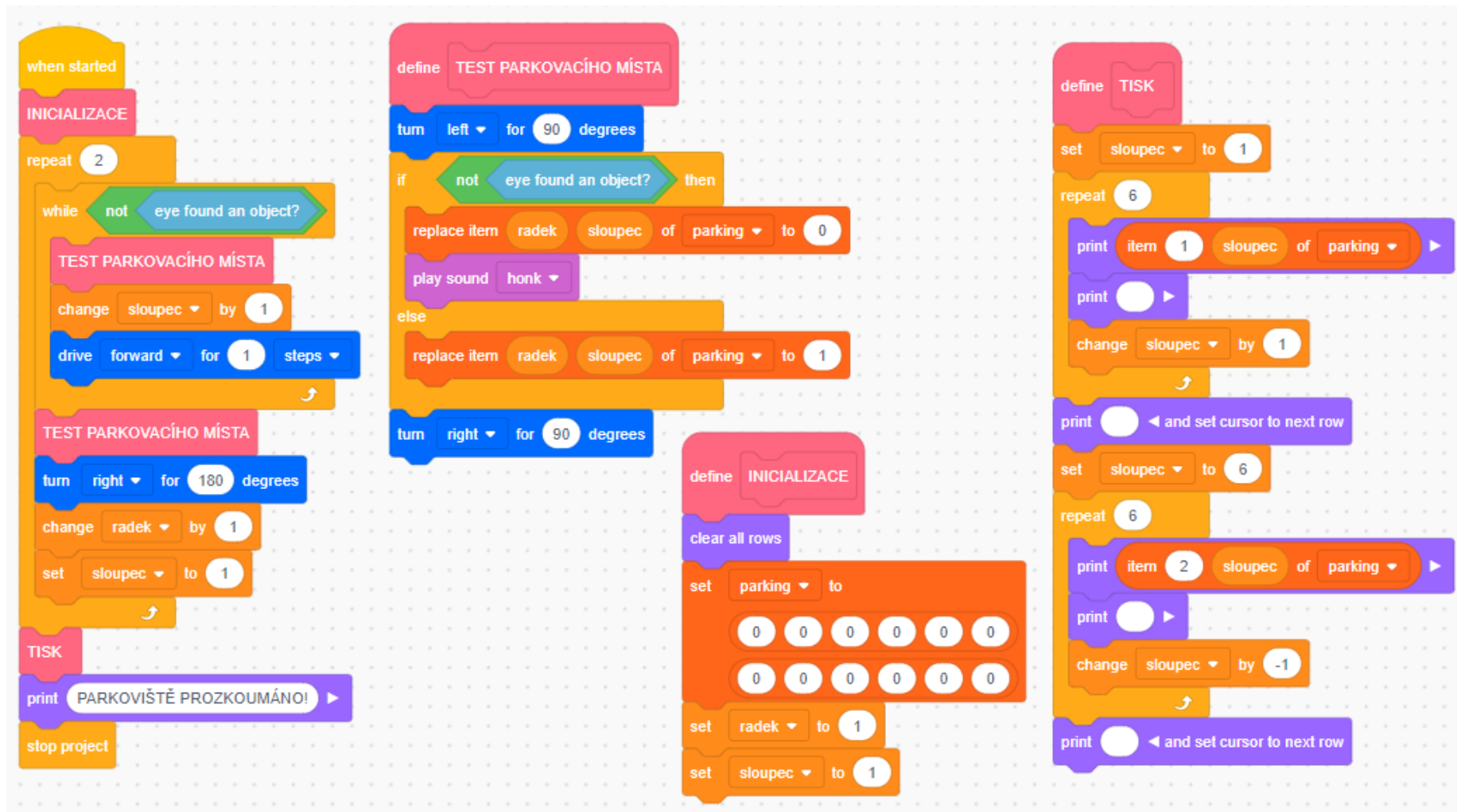
define ODJEZD
  drive forward for 1 steps
  turn left for 90 degrees

define ZAPARKOVÁNÍ
  drive forward for 1 steps
  turn right for 90 degrees
  turn right for 90 degrees

define TEST PARKOVACÍHO MÍSTA
  turn left for 90 degrees
  if not eye found an object? then
    ZAPARKOVÁNÍ
    play sound honk
    ODJEZD
  else
    turn right for 90 degrees
    replace item pozice of parkoviste to 1

define TISK
  set pozice to 1
  repeat 6
    print item pozice of parkoviste
    print 
    change pozice by 1
  print 
  and set cursor to next row
  set pozice to 12
  repeat 6
    print item pozice of parkoviste
    print 
    change pozice by -1
  print 
  and set cursor to next row

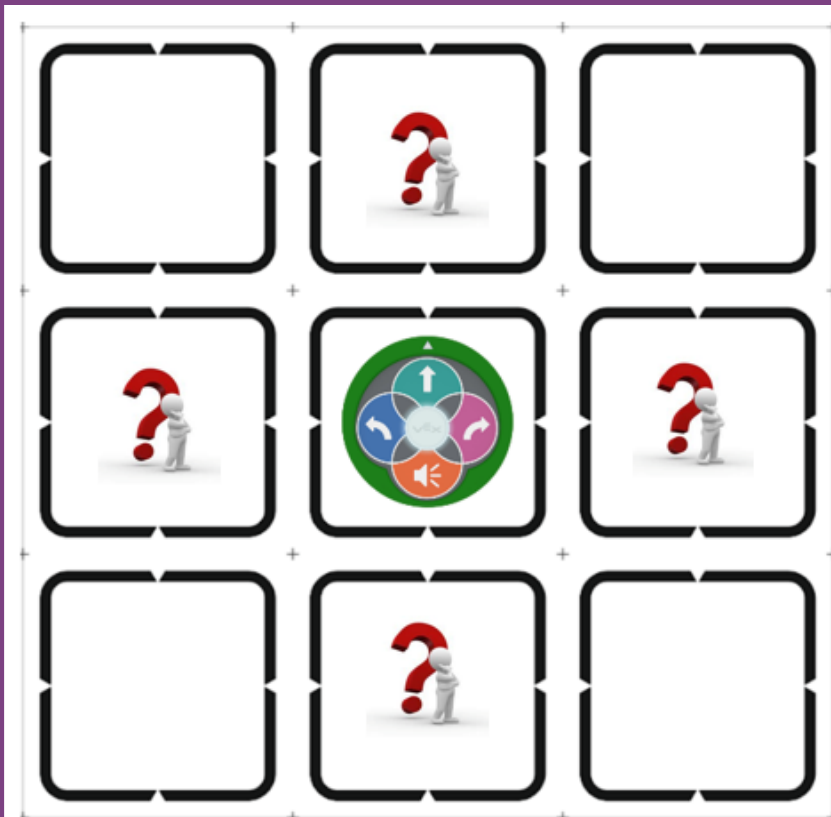
```



Metodická doporučení:

- stejná úloha řešená pomocí dvourozměrného seznamu – žáci by sami měli určit, které z obou řešení je vhodnější

Počítání barev - 1



Vytvoř program, s jehož pomocí VEXík postupně spočítá počet barevných kostiček ve svém okolí. Nejprve vše vyzkoušej na desce s maketou a program zapiš pomocí blokových příkazů do jednoho scénáře. Nakonec vyzkoušej VEXíka naprogramovat a všechny úkoly splň s ním.

Zobrazení výsledků bude probíhat v informačním okně systému code123.vex.com.

```

when started
  clear all rows
  set poradi to 1
  set objekt to 0 0 0 0 0 0 0 0 0
  repeat 8
    test_barev
    turn right for 45 degrees
    change poradi by 1
  vypis
  stop project

define test_barev
  if eye detects green ? then
    replace item poradi of objekt to 1
  else if eye detects red ? then
    replace item poradi of objekt to 2
  else if eye detects blue ? then
    replace item poradi of objekt to 3
  else
    replace item poradi of objekt to 4

define vypis
  set poradi to 1
  repeat 8
    if item poradi of objekt = 1 then
      vypis_zelena
    else if item poradi of objekt = 2 then
      vypis_cervena
    else if item poradi of objekt = 3 then
      vypis_modra
    else
      vypis_bezbarvy
    change poradi by 1

define vypis_bezbarvy
  print poradi
  print . blok - bez barvy and set cursor to next row

define vypis_zelena
  print poradi
  print . blok - zeleny and set cursor to next row

define vypis_modra
  print poradi
  print . blok - modry and set cursor to next row

define vypis_cervena
  print poradi
  print . blok - cerveny and set cursor to next row

```

Počítání barev - 2



Vytvoř program, s jehož pomocí VEXík postupně spočítá počet barevných kostiček ve svém okolí. Nejprve vše vyzkoušej na desce s maketou a program zapiš pomocí blokových příkazů do jednoho scénáře. Nakonec vyzkoušej VEXíka naprogramovat a všechny úkoly splň s ním.

Zobrazení výsledků bude probíhat v informačním okně systému code123.vex.com.


```

when started
  clear all rows
  set rada to 0
  set sloupec to 0
  set kostky to
    0 0
    0 0
    0 0
  pruzkum
  vystup
  stop project

define pruzkum
  repeat 3
    drive forward for 1 steps
    change rada by 1
    set sloupec to 1
    turn left for 90 degrees
    test_kostky
    set sloupec to 2
    turn right for 180 degrees
    test_kostky
    turn left for 90 degrees

define test_cervene
  if eye detects red ? then
    replace item rada sloupec of kostky to 1

define test_zelene
  if eye detects green ? then
    replace item rada sloupec of kostky to 2

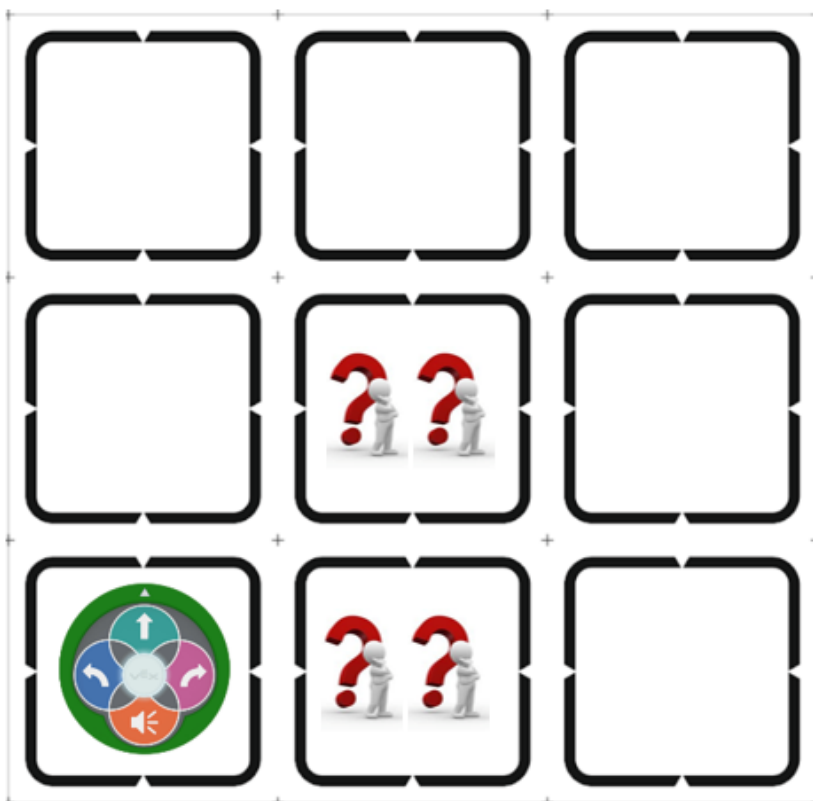
define test_modre
  if eye detects blue ? then
    replace item rada sloupec of kostky to 3

define test_kostky
  if eye found an object? then
    test_cervene
    test_modre
    test_zelene
    test_bezbarvych

define vystup
  print VEXcode

define test_bezbarvych
  if not eye detects green ? or eye detects red ? or eye detects blue ? then
    replace item rada sloupec of kostky to 4
  
```

Počítání barev - 3











Vytvoř program, s jehož pomocí VEXík postupně spočítá počet barevných kostiček ve svém okolí. Nejprve vše vyzkoušej na desce s maketou a program zapiš pomocí blokových příkazů do jednoho scénáře. Nakonec vyzkoušej VEXíka naprogramovat a všechny úkoly splň s ním.

Zobrazení výsledků bude probíhat formou blikání tlačítka VEX na VEXíkovi po stisku tlačítka KLAKSON.

Upravte řešení na libovolný počet vyhodnocovaných řad.

Počítání barev 4


objekt – kvádr

Metodická doporučení:

- poslední úlohy již záměrně uvádím bez řešení, pokud jste se úspěšně „prokousali“ až sem, neměl by být velký problém obě zvládnout i bez dalších pomůcek z mé strany 😊

DOSLOV

Vážené kolegyně, vážení kolegové, právě jste dorazili na úplný konec sbírky učebních materiálů, které jsem sestavil pro výuku algoritmizace a programování v rámci tzv. „nové“ informatiky.

VEX 123 je dle mého názoru skvěle využitelná programovatelná učební pomůcka pro výuku na 1. i 2. stupni ZŠ. Věřím a na základě spolupráce s Mgr. Klárou Tesařovou z Gymnázia a Obchodní akademie v Mariánských Lázních si dokonce troufám konstatovat, že je výzvou i pro studenty víceletých gymnázií.

Sbírka, kterou máte tímto k dispozici, není žádný oficiální učební materiál. Jedná se o výtvar nadšeného učitele informatiky, který se rozhodl své nápady sdílet se svými kolegyněmi a kolegy tak, aby jim usnadnil vstup do výuky informatiky dle revize z roku 2021. Jako takový budu mít radost, pokud se můj výtvar mezi vyučujícími rozšíří a bude plnit svou hlavní funkci – sborník nápadů pro nadšené uživatele VEXíků.

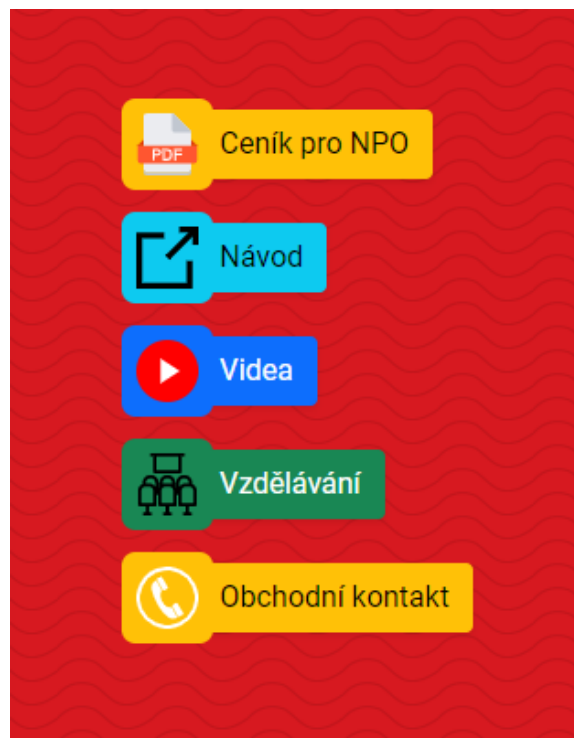
Rád bych ale také získal nějakou zpětnou odezvu, jak jsou mé nápady využívány. Budu vděčný za jakékoliv pochvalné i kritické připomínky, sbírka určitě není uzavřený produkt a může se v čase dále rozvíjet. Stejně tak by bylo velmi pozitivní, kdybyste někdo z vás byl schopen a ochoten se podělit o své nápady a pracovní materiály k využívání VEXíků. Jakékoliv vaše reakce směřujte prosím laskavě na mou školní e-mailovou adresu ict@zskamenna.cz. Potěší mne i samotné oznámení z vaší strany, že jste sbírku začali používat, s radostí budu sledovat, jak se mé nápady a „náказа“ VEXíkem šíří napříč Českem (případně i Slovenskem).

Přeji všem hodně zábavy a zdaru při zařazování prvků robotiky VEX 123 do výuky informatiky.

© Mgr. Milan Šatra, ZŠ Aš, Kamenná 152, 2022

Užitečné odkazy

<https://code123.vex.com/>



www.vexrobotika.cz

www.veskole.cz

Sbírka vznikla s podporou společnosti AV MEDIA SYSTEMS

