Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

DIPLOMOVÁ PRÁCE

Metodické materiály pro výuku robotiky s využitím Lego Mindstorms

2020

Bc. Antonín Vlach

Vedoucí práce: doc. RNDr. Miroslav Kolařík, Ph.D.

Studijní obor: Učitelství výpočetní techniky pro střední školy, prezenční forma

Bibliografické údaje

Autor:	Bc. Antonín Vlach
Název práce:	Metodické materiály pro výuku robotiky s využitím Lego Mindstorms
Typ práce:	diplomová práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2020
Studijní obor:	Učitelství výpočetní techniky pro střední školy, prezenční forma
Vedoucí práce:	doc. RNDr. Miroslav Kolařík, Ph.D.
Počet stran:	68
Přílohy:	1 CD
Jazyk práce:	český

Bibliograhic info

Author:	Bc. Antonín Vlach
Title:	Methodological material on teaching robotics using Lego Mindstorms
Thesis type:	master thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2020
Study field:	Computer Science for Education, full-time form
Supervisor:	doc. RNDr. Miroslav Kolařík, Ph.D.
Page count:	68
Supplements:	1 CD
Thesis language:	Czech

Anotace

V této práci byl vytvořen pracovní materiál pro výuku algoritmizace a robotiky s využitím Lego Mindstorms. Práce popisuje stavebnici Mindstorms a obsahuje materiály k výuce. V textu jsou poskytnuty pracovní listy ve verzích pro učitele a pro žáky doplněné o vzorová řešení úloh v podobě zdrojových kódů. Celkem je takto zpracováno třicet hodin výuky. Vytvořený materiál může učiteli usnadnit přípravu i průběh vyučovacích hodin.

Synopsis

In this thesis, there was created material on teaching algorithmization and robotics using Lego Mindstorms. The thesis describes Mindstorms and offers teaching material. In the text, you can find worksheets for a teacher and for pupils together with sample solutions of tasks as source codes. There is material for thirty lessons. The material helps a teacher with both preparation and execution of lessons.

Klíčová slova: Lego; Mindstorms; výuka; výukové materiály; pracovní listy; robotika; algoritmizace

Keywords: Lego; Mindstorms; teaching; teaching material; worksheets; robotics; algorithmization

V první řadě děkuji vedoucímu práce doc. Kolaříkovi za důvěru, kterou ve mne vložil, i za mnohé věcné připomínky. Dále bych rád poděkoval přátelům a především celé rodině za podporu v průběhu tvorby této práce. V neposlední řadě děkuji dámám Ing. Ludmile Vlachové a Jitce Uhnavé za pomoc s jazykovou a stylistickou korekturou textu.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce podpis autora

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Tato práce zpracovává výuku robotiky s využitím robotické stavebnice Mindstorms značky Lego. Zvláštní důraz při práci je kladen na výuku programování a algoritmizace a podporu algoritmického myšlení žáků. Výsledkem práce je kompletní materiál pro výuku kroužku robotiky na dobu přibližně jednoho školního roku. Výukový materiál obsahuje učitelské pracovní listy, v nichž jsou uvedeny informace potřebné k přípravě a realizaci výuky, a žákovské pracovní listy s úkoly, které mají žáci v hodině řešit. Pracovní listy jsou doplněné vzorovými zdrojovými kódy uvedených úloh.

Výuka robotiky je v současné době populární záležitostí, robotické stavebnice je možné pro školu zakoupit například s využitím různých grantů, často však na škole chybí osoba kvalifikovaná pro tuto výuku. Na některých menších základních školách může dokonce chybět i aprobovaný vyučující informatiky. Cílem této práce je tedy přehledně a jednoduše zpracovat informace o Mindstorms a hlavně poskytnout hotové pracovní materiály, které i méně zkušeného učitele provedou výukou robotiky krok za krokem.

Práce má dvě hlavní části. V první části jsou podrobně rozebrány možnosti, které nabízí stavebnice Lego Mindstorms. Věnuje se krátce historii i současné nabídce různých verzí Mindstorms. Více prostoru však věnuje možnostem, které stavebnice nabízí, popisu jednotlivých robotických součástek a programování robotů. Kapitola může poskytnout užitečné informace každému, kdo s Mindstorms nemá žádné nebo minimální zkušenosti, obsahuje však i některé zajímavé technické údaje o jednotlivých obsažených součástkách.

Největší prostor práce je věnován druhé hlavní části. Tato část uvádí konkrétní výukové materiály v podobě učitelských pracovních listů, které se mohou stát pro učitele průvodcem pro přípravu a realizaci hodiny. Pracovní list pro každou vyučovací hodinu obsahuje cíl výuky, potřebné znalosti, kterými by měl žák disponovat, aby danou hodinu zvládl, dále potřebné pomůcky, zadání úkolů pro žáky a nakonec též komentář s praktickými poznámkami umožňující hladký průběh hodiny i její přípravy. Samostatné pracovní listy pro učitele i žáky a vzorové zdrojové kódy je možné najít v přílohách.

Obrázek 1: Řídicí jednotky různých verzí Mindstorms. Zleva: RCX, NXT, EV3.

1 Lego Mindstorms

Mindstorms je robotická stavebnice značky Lego. Stavebnice Lego Mindstorms je určená k výuce programování nejen (ale zvláště) pro žáky 2. stupně ZŠ a odpovídajících ročníků víceletých gymnázií. Mindstorms je k dostání na e-shopu značky Lego[1], kde je možné zakoupit celé sady nebo jednotlivé součástky.

Od svého vzniku v roce 1998 stavebnice Mindstorms prošla značným vývojem a několika hlavními verzemi (pro porovnání viz obrázek 1):

- Mindstorms RCX (1998),
- Mindstorms NXT (2006),
- Mindstorms EV3 (2013).

V této práci se budeme zabývat verzí EV3, která je nejnovější a v současné době také nejrozšířenější. Nicméně stále se ještě na některých místech může používat starší verze NXT. Funkcionalita prvků verze NXT se však od verze EV3 nijak zvlášť neliší, takže většinu uvedených informací je možné aplikovat na obě verze (NXT i EV3). Dokonce je možné některé (ne však všechny) prvky obou verzí kombinovat. Software k programování verze NXT se liší od novější verze EV3, nicméně roboty verze NXT lze programovat i s použitím aplikace pro EV3. Zdrojové kódy uvedené v této práci budou tedy pro verzi EV3, ale je možné je použít i pro starší NXT. Jediným omezením NXT tak bude absence některých senzorů, které se v EV3 vyskytují a budeme je používat v této práci. [1]

Lego Mindstorms EV3 je k dostání v několika sadách, a to zejména:

- Lego Mindstorms EV3 (31313),
- Lego Mindstorms Education EV3 Core Set (45544),
- Lego Mindstorms Education EV3 Expansion Set (45560).

[1] https://www.lego.com/cs-cz/themes/mindstorms

Každá sada obsahuje různé robotické součástky a dále mnoho stavebních prvků. Funkce stavebních prvků je zřejmá, proto následující části textu popisují pouze jednotlivé robotické součástky a jejich funkci. Mezi robotické součástky patří hlavní řídicí jednotka (tzv. programovatelná kostka), dále pak různé senzory a motory. Všechny tyto části jsou propojeny speciálními kabely, které jsou v základních sadách obsaženy. Kabely jsou univerzální se stejnými konektory pro připojení všech senzorů i motorů. V každém balení jsou k dispozici kabely několika různých délek.

Obě základní sady (31313 a 45544) obsahují řídicí jednotku a stejné motory. Konstrukční prvky jsou v každé sadě různé. Výskyt senzorů se v jednotlivých základních sadách mírně liší (více v kapitole 1.2.2). Sady se liší i v samotném balení. Zatímco obě sady pro vzdělávání (základní i rozšiřující) jsou uloženy v plastových boxech obsahujících mezipatro pro uspořádání stavebních dílků, sada 31313 je dodávána pouze v papírové krabici. Řídicí jednotka může být napájena bateriemi typu AA nebo akumulátorem. Ten je obsažen pouze v balení pro vzdělávání (45544), je však možné jej dokoupit samostatně. Software i hardware ve verzi pro vzdělávání (45544) skýtá některé funkce, které klasická základní sada (31313) neumožňuje (např. logování měřených hodnot, úpravu pokynů a zobrazení některých úloh). [2]

Rozšiřující sada pro vzdělávání (45560) neobsahuje robotické součástky, ty jsou již obsažené v sadě základní (45544). Rozšiřující sada obsahuje více stavebních součástek umožňujících konstrukci větších a sofistikovanějších robotů.

V tomto textu budeme předpokládat práci se sadou pro vzdělávání, a to obvykle pouze se sadou základní (45544). Velká část textu jistě bude platit univerzálně i pro klasickou sadu (31313), v takovém případě je však třeba postupovat opatrně a vždy ověřit dostupnost potřebných prvků. Stejný postup doporučujeme i u dalších zdrojů informací a úloh na toto téma.

Obrázky a některé informace uvedené v následujících oddílech byly převzaty z oficiálního e-shopu značky Lego (odkaz výše).

1.1 Programovatelná kostka

Hlavní řídicí jednotka stavebnice Lego Mindstorms se nazývá programovatelná kostka, někdy též inteligentní kostka, případně jen kostka. Je hlavním prvkem každého robotu. Na programovatelnou kostku se dále napojují všechny senzorické, motorické i stavební součástky.

Kostka disponuje procesorem s architekturou ARM9 a výpočetní rychlostí 300 MHz. Na procesoru běží operační systém s linuxovým jádrem. K uložení programů a jejich dat slouží paměti Flash o velikosti 16 MB a RAM o velikosti 64 MB. Dále je kostka vybavena monochromatickým LCD displejem s rozlišením 178×128 pixelů. Kostku je možné ovládat pomocí šesti tlačítek (čtyř navigačních a dvou pro potvrzení a zrušení). Pod navigačními tlačítky se nachází signální světlo, které může svítit třemi různými barvami (zelená, oranžová, červená). [3]

Kostku v pohledu shora ukazuje obrázek 2. Na zadní straně (za displejem) je

Obrázek 2: Programovatelná kostka v pohledu shora zleva.

umístěn USB port (typ B mini) sloužící k propojení kostky s počítačem. Zezadu má kostka též 4 porty označené A–D, ke kterým je možné připojit robotické motory. Na protilehlé (přední) straně se nacházejí podobné 4 porty označené 1–4, které slouží k připojení robotických senzorů. Na levé straně kostky se nachází USB port, který umožňuje propojovat více kostek nebo připojit ke kostce nějaký externí modul (např. Wi-Fi modul). Vedle USB portu je slot pro kartu microSD, kterou lze použít například pro navýšení paměti. Paměťová karta může také obsahovat novější, vylepšený nebo upravený firmware. Zprava můžeme na kostce vidět průduchy pro reproduktor, kterým kostka vydává zvukové signály nebo přehrává nahrávky přidané k programu. Na spodní straně se nachází baterie. Kostku je možné napájet ze speciálního lithium-iontového akumulátoru o kapacitě 2050 mAh. Kostka může být napájena také šesti bateriemi typu AA.

1.1.1 Vlastní uživatelské rozhraní kostky

Samotná kostka má vlastní uživatelské rozhraní využívající LCD displej a tlačítka. Uživatelské rozhraní kostky skýtá samo o sobě mnoho možností. Za běhu programu mohou být některé informace zobrazeny na LCD displeji, ať už v podobě textu, nebo obrazu. Robot může přijímat instrukce použitím pěti navigačních tlačítek. Šesté tlačítko zastaví běžící program. Pokud program neběží, nabízí uživatelské rozhraní další řadu možností. Můžeme z něj sledovat připojené senzory a přímo ovládat připojené motory. Dále můžeme vybírat a spouštět různé programy, které byly do kostky nahrány dříve. Navíc je možné z uživatelského rozhraní kostky provádět řadu nastavení. Běžící uživatelské prostředí kostky na displeji ukazuje obrázek 3.

Samotné uživatelské prostředí se skládá ze 4 karet. Z první karty je možné spouštět předem nainstalované programy, které byly spuštěny nedávno. Na druhé kartě lze spravovat uložené soubory, tedy hlavně programy (ať už jsou uloženy ve formě binární nebo ve formě zdrojového kódu) a jejich pomocné soubory (obrázky, zvuky). Třetí karta nabízí správu senzorů a motorů. Je možné zde sledovat všechny porty a vše, co je k nim připojeno. Pokud je k danému portu připojen

Obrázek 3: Uživatelské prostředí kostky.

nějaký senzor, je možné číst aktuální naměřené hodnoty. Pokud jsou ke kostce připojeny nějaké motory, lze je přímo spustit pomocí tlačítek na kostce. Dále je možné dokonce částečně upravovat uložené programy, avšak pouze ty, které jsou uloženy ve formě zdrojového kódu, nikoli jako binární soubory. Na čtvrté kartě se nachází nastavení. Zde je možné upravit hlasitost zvuku a dobu, po které kostka přejde do režimu spánku. Můžeme zde najít i nastavení Bluetooth a Wi-Fi nebo vyhledat informace o kostce, jako například verze firmware či operačního systému, výrobní číslo, volnou kapacitu paměti a další.

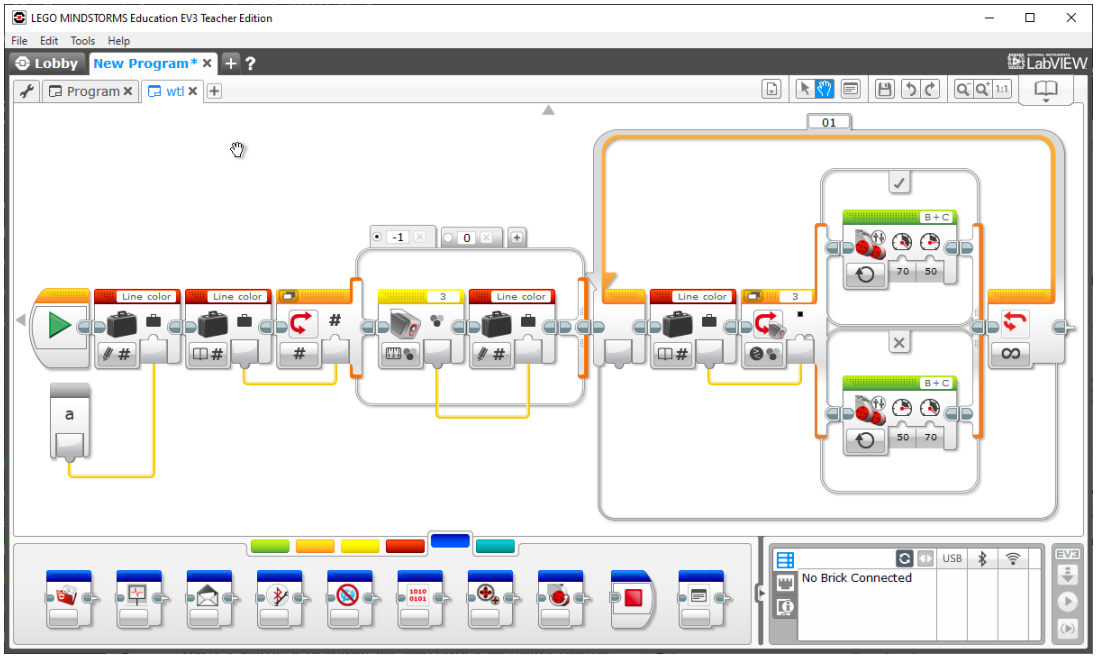
1.1.2 Propojení a programování

Kostka a potažmo celý robot může být programován z mnoha různých zařízení. Aplikace sloužící pro připojení robotu a jeho naprogramování je k dostání pro systémy Windows (7 a novější), MacOS, iPad, Android (tablet) a mnohé další. Je možné ji zdarma stáhnout na webu Lego Education[2] nebo budete přesměrováni do příslušného internetového obchodu podle daného operačního systému.

Kostku se zařízením lze propojit několika způsoby. Nejjistější je propojení přes USB kabel, který bývá obvykle součástí balení. Výhodou připojení přes USB je rychlé navázání spojení bez nutnosti cokoli nastavovat. Spojení je po celou dobu stabilní, samozřejmě pokud je USB kabel i jeho konektory kvalitní a v dobrém stavu. Nevýhodou však je omezená délka kabelu. Obvykle chceme hotový program rovnou vyzkoušet. Pokud však program zahrnuje pohyb robotu, bude nutnost připojeného kabelu značnou komplikací.

Odstranění této nevýhody nabízejí další dva bezdrátové způsoby připojení: přes Bluetooth nebo Wi-Fi. Připojení přes Bluetooth, jakmile je navázáno, bývá poměrně stabilní při zachování volnosti pohybu. Nicméně občas bývá náročnější spojení navázat. Poprvé je nutné spojení navázat (spárovat zařízení) přes uživatelské rozhraní kostky. Při opakovaném připojování dříve spárovaných zařízení se obvykle spojení naváže automaticky nebo pomocí několika kliknutí v aplikaci, občas je však potřeba v uživatelském rozhraní kostky dřívější spojení odstranit a navázat (spárovat) nové od začátku. Spojení přes Wi-Fi nebylo vyzkoušeno,

[2] https://education.lego.com/en-us/downloads/mindstorms-ev3/software



Obrázek 4: Programování Mindstorms pomocí bloků.

neboť je pro něj potřeba externí Wi-Fi modul připojený do USB portu, což je samo o sobě značná nevýhoda.

Je výhodné připojit kostku k počítači (resp. tabletu aj.) ještě, než začneme programovat. Kostka umí sama identifikovat většinu připojených motorů a senzorů. Aplikace pak může rovnou při vytváření programu automaticky nastavit čísla (resp. písmena) portů, kam jsou motory či senzory připojeny. Jinak (bez připojení) program zvolí výchozí porty a pravděpodobně je pak bude nutné ručně nastavovat podle zapojení našeho robotu. Když máme hotový robot připojený už v době programování, můžeme program ladit po částech a pomocí aplikace sledovat hodnoty čtené na senzorech.

Příkazy pro Mindstorms mají podobu bloků, které se přetažením skládají za sebe. Každý blok může dále obsahovat specifická nastavení různých parametrů. Základní sada obsahuje bloky pro čtení senzorů, ovládání motorů, běžné programové konstrukce, práci s proměnnými a daty, posílání zpráv a další. Dále je možné vytvářet vlastní nové bloky poskládáním bloků existujících a nastavením vstupních i výstupních parametrů nového bloku. Vzhled aplikace při programování pomocí bloků ukazuje obrázek 4. V horní části můžeme přepínat mezi více programy nebo podprogramy. V dolní části se nachází nabídka příkazů (bloků). Většinu plochy (uprostřed) pak zabírá samotný zdrojový kód programu.

Aplikaci je možné nainstalovat ve verzi pro studenta a pro učitele. Verze pro učitele obsahuje některé volitelné materiály navíc, ale jinak se v zásadě neliší. Aplikace umožňuje propojit robot s počítačem (resp. tabletem), vytvářet a nahrávat do robotu nové programy i ladit programy za běhu. Dále aplikace nabízí řadu tutoriálů zpřístupňujících všechny nabízené základní možnosti nebo něko-

lik návodů pro sestavení různých robotů. Kromě toho je možné aplikaci využít k logování a zpracování dat naměřených a odeslaných robotem v průběhu experimentu.

Mindstorms je možné programovat také v jazyce Python. Je k tomu zapotřebí microSD karta, na kterou se nahraje uzpůsobený firmware. Dále je třeba do počítače nainstalovat Visual Studio Code editor a do něj příslušné rozšíření pro programování Mindstorms. Pro začátečníka je tento postup jistě zcela nevhodný, může být však užitečný pro pokročilejší práci s Mindstorms. [4] Mindstorms je možné programovat i v jiných jazycích, ale je třeba použít příslušně upravený firmware a software třetích stran. Firmware je možné libovolně upravovat právě díky jeho linuxovému jádru.

1.2 Robotické součástky

Celý robot se skládá z mnoha konstrukčních dílů. Jedná se o klasické součástky stavebnice Lego. Mozkem robotu je programovatelná kostka, které se podrobně věnuje kapitola 1.1. To, co dává robotu život, jsou robotické součástky. Jedná se o motory, které umožňují robotu pohyb, a senzory, kterými robot vnímá okolní svět. Následující části textu popisují motory a nejběžnější senzory a ukazují jejich použití v programu.

1.2.1 Motory

Lego Mindstorms má dva různé typy motorů: velký a střední. Jak motory vypadají, můžete vidět na obrázku 5.

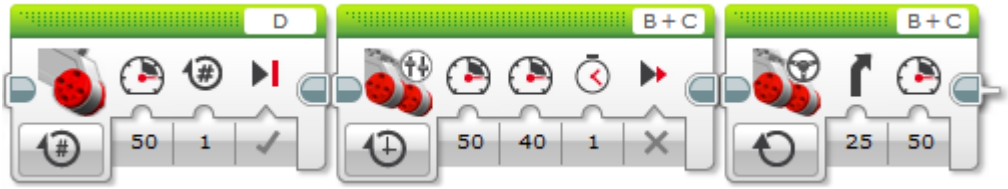
Obrázek 5: Dva typy motorů: velký (vlevo) a střední (vpravo).

Velký motor je silný motor, který je obvykle používán jako hlavní pohonná jednotka. Může pohánět třeba kola nebo pásy. Obvykle se používá kombinace dvou motorů, na každé straně vozidla jeden, což umožňuje snadné zatáčení. Velký motor zvládá maximální rychlost přibližně 160–170 otáček za minutu. Jeho otáčivý moment je 20 N·cm. V motoru je zabudován senzor otočení, který měří s přesností na 1 ° a umožňuje tak otočit motor o konkrétní zadaný úhel či počet otáček.

Střední motor je slabší a používá se obvykle k pohonu menších pohyblivých částí robotu (např. zdvižná ramena, otáčení senzorů aj.). Točivý moment motoru je pouze přibližně 8 N·cm, nicméně při nízkém zatížení je schopen otáčet se až

rychlostí 240–250 otáček za minutu, což je mnohem rychleji, než zvládá velký motor. Stejně jako velký motor je i střední motor vybaven senzorem otočení, který měří s přesností na 1 °.

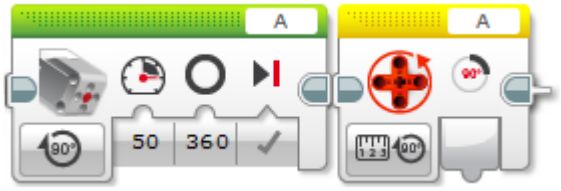
Velký motor lze ovládat několika příkazy. Základní příkaz nabízí zastavit či spustit motor. Spustit jej lze na dobu neurčitou, nebo na určený čas, konkrétní počet otáček či zadaný úhel. Při spouštění motoru lze nastavit rychlost/sílu běhu motoru. Při zastavení motoru je možno vybrat, zda má být motor okamžitě zastaven s použitím brzdy, či zda se nechá volně doběhnout. Další příkaz pojmenovaný „pohyb tanku“ umožňuje ovládat dva motory jedním příkazem. Při použití dvou motorů (umístěných na stranách) jako pohonu vozidla je většina parametrů pohybu pro oba motory stejná. Jediné, co se může lišit (v případě zatáčení), je rychlost jednotlivých motorů. Tento příkaz tedy umožňuje nastavit rychlost každému motoru zvlášť a všechny ostatní parametry oběma motorům najednou. Třetí příkaz nazvaný „pohyb s volantem“ umožňuje zadat společnou rychlost i ostatní parametry oběma motorům. Navíc lze ještě nastavit míru zatočení. Dělá v podstatě to stejné jako předchozí příkaz, pouze skrytě, čímž dosahuje větší intuitivnosti pohybu.



Obrázek 6: Různé příkazy/bloky k ovládání jednoho nebo dvou velkých motorů.

Zmíněné příkazy ukazuje obrázek 6. První příkaz spustí velký motor na portu D na jednu otočku se silou 50 % a poté s použitím brzdy motor zastaví. Druhý příkaz spustí oba motory na portech B a C, levý s výkonem 50 % a pravý s výkonem 40 % na dobu 1 sekundy a poté nechá motory volně doběhnout. Třetí příkaz spustí oba motory se společným výkonem 50 % a zatočením 25 % vpravo na dobu neurčitou, tedy motory poběží se současným nastavením, dokud nebude nastavení změněno, popř. motory úplně zastaveny.

Příkazy týkající se středního motoru ukazuje obrázek 7. Příkaz k ovládání středního motoru funguje v podstatě stejně jako příkaz k ovládání jednoho velkého motoru. Uvedený příklad spustí střední motor na portu A s výkonem 50 %, otočí motor o 360 ° a poté okamžitě zastaví s použitím brzdy. Za zmínku stojí hlavně druhý blok ukázaný na obrázku 7. Je to příkaz, který pouze čte hodnotu senzoru otočení umístěného v motoru. Sám o sobě nebude mít příkaz na činnost robotu žádný efekt, ale získanou hodnotu je možné použít v dalších příkazech nebo při dalším rozhodování. Senzor umožňuje získat hodnotu ve stupních nebo počtech otáček. Je možné měřit také okamžitý výkon otáčení. Stejný příkaz lze použít i u velkého motoru.



Obrázek 7: Různé příkazy/bloky k ovládání středního motoru.

1.2.2 Popis senzorů

K programovatelné kostce Mindstorms EV3 je možné připojit celou řadu senzorů, ať už originální od značky Lego, nebo senzory od dalších výrobců, které je možné dokoupit zvlášť. Tato část textu popisuje pět základních senzorů: dotykový, ultrazvukový, barevný, gyroskopický a infračervený. Tyto vybrané senzory jsou všechny originální od značky Lego a jsou obsaženy alespoň v některých základních sadách (viz kapitolu 1). Sada Mindstorms EV3 (31313) obsahuje senzory dotyku, barev a infračervený senzor. Sada Education EV3 Core Set (45544) obsahuje senzor barev, dotyku, otočení (gyroskopický) a vzdálenosti (ultrazvukový). Všechny zmíněné senzory můžete vidět na obrázku 8.

Obrázek 8: Nejběžnější senzory. Zleva: dotykový, ultrazvukový, barevný/světelný, gyroskopický, infračervený.

Dotykový senzor je nejjednodušší ze všech senzorů. Jedná se v podstatě o tlačítko. Rozlišuje pouze dvě hodnoty: volný, stisknutý. V některých případech ještě program umí rozlišovat hodnotu „kliknuto“, která znamená stisknutí a uvolnění tlačítka po sobě.

Ultrazvukový senzor umí určovat vzdálenost od nejbližšího předmětu. Vysílá ultrazvukové vlny a pomocí echolokace, tedy detekce odražených vln, spočítá vzdálenost předmětu, od kterého se vlna odrazila. Vzdálenost je možné měřit v centimetrech či palcích. Rozsah měřitelných hodnot je 1–255 cm, resp. 1–100 palců. Přesnost měření se udává ± 1 cm, ale při malé vzdálenosti (menší než 5 cm) se přesnost snižuje. Dále senzor umí detekovat jiné zdroje ultrazvuku, kterými mohou být například další roboty Mindstorms.

Barevný senzor v prvním módu dokáže číst barvu předmětu před sebou. Dokáže rozlišit osm různých barev: černá, modrá, červená, žlutá, zelená, bílá, hnědá, neurčitá. Jedná se v podstatě o všechny barvy, které mají různé kostky Lega. Poslední možnost znamená, že žádná z uvedených barev nebyla rozpoznána. V druhém režimu senzor vyzařuje světlo a čte intenzitu světla odraženého, čímž nejlépe

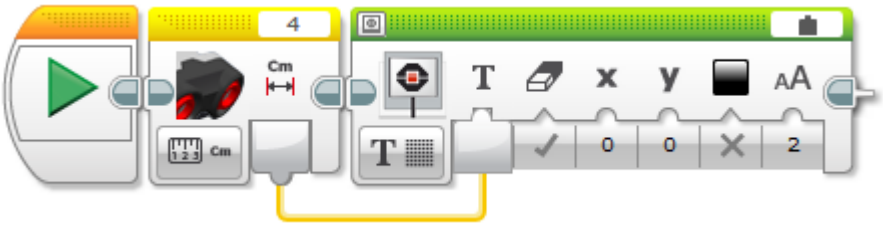
určí světlost materiálu nezávisle na okolním světle. Ve třetím režimu senzor jednoduše detekuje intenzitu světla vyzářeného z okolí (tzv. ambientního). Hodnoty intenzity světla (ať už odraženého nebo ambientního) se udávají v procentech.

Gyroskopický senzor umožňuje měřit úhel otočení. Data dává s přesností ±3 °. Dále umožňuje zjistit aktuální rychlost otáčení ve stupních za sekundu s maximální měřitelnou rychlostí 440 stupňů za sekundu. Při otáčení kolem osy se úhel stále zvyšuje i přes pomyslné hranice 0 ° a 360 °. Omezením gyroskopického senzoru je, že detekuje otáčení pouze podle jedné osy, takže nemůže zajistit plnou orientaci v prostoru. Aby gyroskopický senzor měřil správně, musí být v době připojení ke kostce v klidu.

K infračervenému senzoru patří ještě speciální ovladač. Senzor umožňuje číst stisknutá tlačítka nebo jejich kombinace na dálkovém ovladači a podle toho se rozhodovat. Ovladač i senzor je možné přepínat mezi čtyřmi kanály, tedy můžeme používat naráz až čtyři ovladače například pro čtyři různé roboty nebo rozšířit možnosti jednoho robotu s použitím více ovladačů. Dále může infračervený senzor pracovat v režimu přiblížení, ve kterém odhaduje vzdálenost objektu před sebou podle odraženého infračerveného světla. Funguje však s menším rozsahem i přesností než ultrazvukový senzor.

1.2.3 Použití senzorů

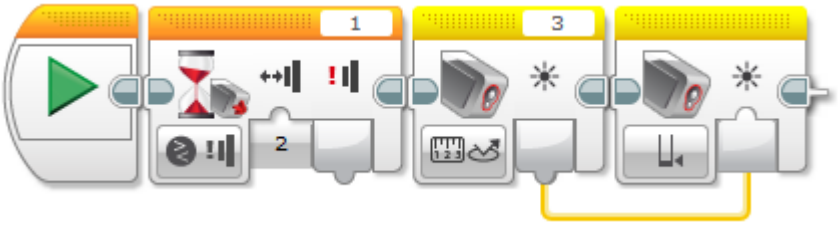
Data ze senzorů se obecně dají využít několika způsoby. Prvním způsobem je příkaz pro jednoduché přečtení dat ze senzoru. Přečtená hodnota poté může být použita v programu jako parametr nějakého jiného příkazu. Příklad prostého přečtení hodnoty ukazuje obrázek 9. Úsek kódu na obrázku přečte vzdálenost v centimetrech pomocí ultrazvukového senzoru a poté ji vypíše na displej.



Obrázek 9: Příklad využití senzoru. Program vypíše naměřenou vzdálenost.

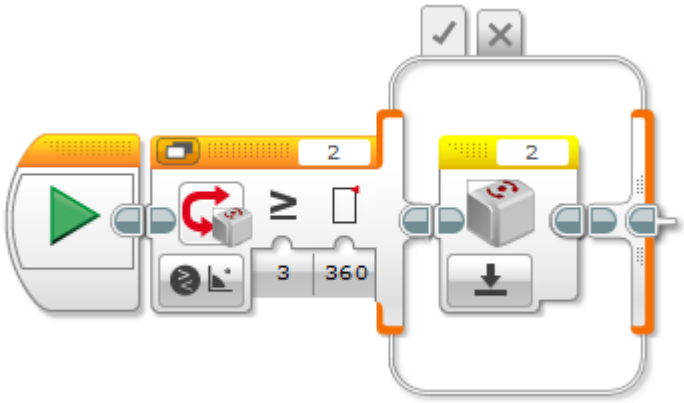
Další možnost využití senzorů v programu nabízí příkaz čekání. Program tak může čekat, dokud senzor nepřečte nějakou konkrétní hodnotu nebo než bude hodnota v zadaném rozsahu. Je možné nastavit i čekání, než se čtená data změní o určitou hodnotu. Využití příkazu čekání v kombinaci s dotykovým senzorem ukazuje obrázek 10. Program kalibruje barevný senzor (viz níže), ale nejprve čeká na kliknutí na tlačítko dotykového senzoru.

Data ze senzorů je možné využívat v příkazech větvení. Nějakou část kódu lze vykonat v závislosti na datech čtených ze senzoru. Příklad podmíněného příkazu



Obrázek 10: Příklad kalibrace spodní hranice odraženého světla po kliknutí na tlačítko.

ukazuje program na obrázku 11, který resetuje gyroskopický senzor, ale pouze v případě, že měří hodnoty vyšší než 360 °.



Obrázek 11: Vynulování gyroskopického senzoru.

Data ze všech senzorů lze využívat podobně. Můžeme čekat na konkrétní hodnotu nebo její změnu, používat naměřené hodnoty při rozhodování nebo jako parametr libovolných příkazů.

Senzor barev dále umožňuje kalibrovat detekci odraženého světla. Je možné mu nastavit referenční minimální a maximální hodnotu, abychom mohli využít plný rozsah hodnot. V reálném světě totiž obvykle černá barva není úplně černá, ale pouze nejtmavší šedá v okolí. Příklad kalibrace ukazuje již dříve zmiňovaný obrázek 10. Program nejprve vyčká na stisknutí tlačítka. Uživatel by měl v tu chvíli namířit barevný senzor na nejtmavší část sledované plochy a poté stisknout tlačítko. Program přečte hodnotu ze senzoru a tu nastaví jako hodnotu minimální.

Gyroskopický senzor je možné resetovat. Při otočení senzoru o více než jednu otáčku měřené hodnoty přesáhnou pomyslné hranice 0 ° a 360 °. Při opakovaném otáčení senzoru kolem jeho osy tak můžeme dostávat extrémně velká a nicneříkající čísla. V takovém případě je užitečné senzor resetovat. Podmíněný reset při přesáhnutí hodnoty 360 ° ukazuje program na obrázku 11. Stejným způsobem je možné vynulovat také senzory otočení vestavěné v motorech.

2 Tvorba výukových materiálů

Hlavním cílem této práce byla tvorba výukových materiálů. Nejprve bylo však nutné definovat kontext, tedy cíle, způsob a podmínky výuky. V této práci vycházíme z předpokladu, že výuka probíhá spíše ve formě zájmového kroužku vzhledem k tomu, že v klasické výuce informatiky na druhém stupni základních škol nebývá algoritmizaci věnováno příliš prostoru. Použití materiálů v klasické výuce však není vyloučeno.

Předmětem výuky je Lego Mindstorms. Výuka zahrnuje sestavení a hlavně programování robotů Mindstorms. Žáci se naučí konstruovat roboty, použít v nich správné součástky a hlavně sestavené roboty programovat. V programech se žáci naučí používat všechny dostupné senzory, číst a zpracovávat čtené hodnoty a na základě těchto hodnot provést další akci, obvykle pomocí robotických servomotorů, doplněných grafickým a zvukovým výstupem.

Výuka by měla probíhat prakticky. To znamená, že každý žák by měl mít možnost programovat samostatně a své programy zkoušet na skutečném robotu. Každou hodinu žák dostane seznam úkolů, které má splnit. Na řešení úkolů může přijít sám, případně mu může učitel nebo spolužák poradit. Nikdy by však neměl žák pouze sledovat, jak někdo řeší úkol za něj. Od každého žáka by tedy měla být vyžadována patřičná samostatná aktivita.

Hlavním cílem výuky nicméně není naučit žáky používat a programovat Lego Mindstorms. Lego Mindstorms je z pedagogického hlediska pouze prostředkem. Hlavním cílem je výuka programování a algoritmizace obecně. Při tvorbě výukových materiálů je tedy kladen důraz na zařazení základních algoritmických prvků. Žák se tak postupně učí používat ve svých programech řídicí konstrukce, jako jsou třeba porovnání, podmínky či cykly. K ukládání informací používá proměnné a později i pole. Program zjednodušuje a zpřehledňuje použitím vlastních příkazů (procedur) s možností použití vstupních i výstupních parametrů. V pokročilejších hodinách se žáci setkají s typickými základními algoritmy, jako je hledání minima či řazení čísel.

Výukové materiály jsou určeny pro žáky druhého stupně základních škol a odpovídajících ročníků víceletých gymnázií. Je zapotřebí vyvinuté formální myšlení, proto by běžní žáci prvního stupně mohli mít s pokročilejšími částmi výuky problémy. Pro pochopení principů a plnění úkolů však nepotřebují žáci žádné speciální vědomosti či dovednosti. Není tedy vyloučeno, aby se nadaní mladší žáci výuky účastnili. Počet žáků je omezen pouze kapacitou učebny a dostupností vybavení. Bylo by dobré, aby každý žák měl k dispozici jednu sadu Mindstorms a jeden počítač. V případě nedostatku vybavení je možné spojovat žáky do skupin, výuka ve skupinách o více než dvou žácích však ztrácí na efektivitě.

Umístění výuky nehraje příliš velkou roli. Některé školy disponují speciální odbornou učebnou pro výuku technických předmětů. Postačí však i klasická počítačová učebna či libovolná jiná učebna. Je však potřeba, aby každý žák (popř. každá skupina žáků) měl přístup k počítači nebo notebooku. Je vhodné, aby počítače byly vybaveny modulem Bluetooth pro snazší nahrávání programů do

robotů. Dále by bylo dobré, aby v učebně byla dostatečně velká volná plocha (ideálně volná část podlahy) pro pokusy s roboty. Některé další potřebné pomůcky či požadavky na prostor jsou uvedeny ve výukových materiálech u jednotlivých hodin.

Vyučující by měl mít alespoň minimální zkušenost s Mindstorms a s programováním. Měl by rozumět zdrojovým kódům programů k jednotlivým úkolům a měl by zvládat programování Mindstorms a ladění programů, aby byl žákům schopen pomoci v případě skryté chyby v jejich programech. Měl by být schopen rozlišit, zda jiný program, než je ten vzorový, může stejně splnit tentýž úkol, a zhodnotit klady a zápory obou řešení.

Výuka je koncipována na lekce trvající 60–90 minut. Potřebný čas ke zvládnutí úkolu se však může výrazně lišit v závislosti na individuálních schopnostech žáků. Navíc výukové materiály doposud nebyly otestovány na reálných žácích. Je proto třeba myslet na časovou flexibilitu. Je možné si vždy zvolit nutné úkoly a rozšiřující. Některé hodiny bude nejspíš třeba doplnit dalšími úkoly nebo naopak rozdělit do více hodin.

Následuje nejprve stručný přehled obsahu výukových materiálů a dále samotné výukové materiály pro jednotlivé hodiny.

2.1 Obsah učiva

Počet hodin je odhadován k pokrytí jednoho školního roku. Vycházíme z předpokladu, že výuka volnočasového kroužku robotiky probíhá jednou týdně. Školní rok má přibližně čtyřicet vyučovacích týdnů. Zájmové kroužky však obvykle nezačínají hned v prvním týdnu výuky a končí o něco dříve. Výukové materiály jsou zde uvedeny na třiceti pracovních listech. Některé pracovní listy obsahují výuku na několik hodin. Metodické materiály v této práci uvedené tedy vydají na celý školní rok.

Obsah výuky je zaměřen na seznámení s činností a použití jednotlivých částí Mindstorms a na základy algoritmizace. Pořadí jednotlivých hodin, jak jsou uvedeny, není náhodné a je vhodné jej při výuce dodržovat. Při sestavení materiálu bylo dbáno na vzájemnou návaznost učiva jednotlivých hodin a věnována pozornost tomu, aby hodina zaměřená na poznání nového konstruktu vždy předcházela hodinám, ve kterých se daný konstrukt využívá. Hodiny jsou řazeny tak, aby související části byly probrány po sobě a aby náročnější hodiny byly proloženy hodinami jednoduššími či přípravnými.

Na začátku jsou uvedeny hodiny zaměřené na poznání možností Mindstorms. Žáci se seznamují s ovládáním motorů, čtením dat pomocí senzorů a použitím vstupu a výstupu programovatelné kostky. Další hodiny cílí na výuku základů algoritmizace. Ukazují použití podmínek, cyklů, proměnných, polí či vlastních podprogramů. V závěru se nachází několik hodin zaměřených na komplexní řešení s využitím dříve naučených vědomostí a dovedností.

2.2 Jednotlivé pracovní listy

Tato část textu uvádí výukové materiály pro jednotlivé hodiny. Výukové materiály jsou uvedeny v podobě pracovních listů. Cílem této práce bylo vytvořit pracovní listy pro učitele i pro žáky.

Učitelské pracovní listy obsahují cíl výuky a popisují potřebné předchozí znalosti, kterými by žáci měli disponovat, aby hodinu zvládli. Dále je zde uveden seznam pomůcek potřebných pro realizaci dané hodiny. Hlavní částí pracovního listu je seznam úkolů, které mají žáci vypracovat. Každý pracovní list uzavírá komentář obsahující metodické poznámky k efektivnějšímu provedení vyučovací hodiny. Následující části této kapitoly jsou právě tyto učitelské listy.

Žákovské listy obsahují pouze úkoly pro žáky, aby mohli žáci pracovat samostatně. Všechny informace obsažené v žákovském listu je tedy možné najít i v listu učitelském. Z toho důvodu v této kapitole žákovské listy nejsou uvedeny. Samostatné žákovské pracovní listy i samostatné listy pro učitele jsou k dispozici v příloze. Nejdůležitější přílohou jsou však vzorové zdrojové kódy. Jedná se o vzorová řešení jednotlivých úloh. Ty nejsou řešeny v učitelských listech, neboť si učitel může otevřít přiložené zdrojové kódy přímo v aplikaci Mindstorms ve svém počítači.

2.2.1 Úvodní hodina

Cíl hodiny:

Žák se seznámí se stavebnicí Mindstorms, prostředím kostky a programovací aplikací. Zvládne vytvořit nový program a nahrát ho do kostky.

Předchozí znalosti:

Jedná se o úvodní hodinu, nepředpokládají se žádné předchozí znalosti.

Potřebné pomůcky:

- Programovatelná kostka
- Jeden střední (nebo velký) motor
- Senzor dotyku a senzor barev/světla
- 1–3 kabely na propojení
- USB kabel k propojení počítače a kostky

Úkoly:

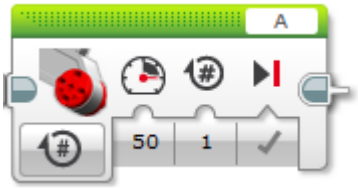
1. Spusťte kostku.

2. Připojte libovolný motor (port A), senzor dotyku (port 1), senzor barev/světla (port 3).

3. Na třetí kartě v prostředí kostky sledujte hodnoty měřené senzory a zkuste ovládat motor.

4. Spusťte aplikaci, vytvořte nový projekt/program a připojte kostku k PC.

5. Vytvořte program, který otočí motorem (viz obrázek 12).



Obrázek 12: Příkaz k ovládání jednoho motoru.

6. Vyzkoušejte různé možnosti ovládání motoru. Vyzkoušejte různé druhy motoru (velký i střední).

7. V pravé dolní části obrazovky si otevřete Port View a sledujte hodnoty motorů a senzorů (viz obrázek 13).

Obrázek 13: Zobrazení hodnot na jednotlivých portech, tzv. Port View. Nachází se v pravém dolním rohu aplikace.

8. Pokuste se zjistit, jaké všechny hodnoty umí připojené senzory číst.

Komentář:

V úvodní hodině je hlavním cílem seznámení žáků s prostředím kostky i aplikace. Proto není třeba sestavovat žádný robot, čímž by se ztrácelo zbytečně mnoho času. Pro první hodinu postačí spíše sada snadných úkolů. Uvidíte sami, jak si žáci s úkoly poradí. Nechte žákům volnost v experimentování. Můžeme předpokládat, že na první hodině budou žáci zvědaví a budou chtít vyzkoušet různé možnosti sami.

Očekávejte zvídavé dotazy žáků typu „K čemu je toto?“ nebo „Co se stane, když udělám tohle?“. Na podobné otázky se nabízí jednoduchá odpověď: „Zkuste to a uvidíte.“ Z vlastního pokusu si žáci odnesou nejvíce nových poznatků. Než však žákům takovou odpověď dáte, pečlivě zvažte případná rizika. Každý pokus musí být bezpečný jak pro žáky, tak pro vybavení, aby nedošlo ke zranění nebo k poškození pomůcek.

Pro zachování bezpečnosti je třeba nastavit pravidla, za jakých bude celá výuka probíhat. Tento krok je vhodné provést ihned na začátku hodiny, ještě než žáci dostanou stavebnice do rukou. Kromě nastavení pravidel je možné žákům krátce popsat jednotlivé důležité části Mindstorms. Součástí této ukázky a stanovení pravidel by mohlo být vysvětlení organizace dílků stavebnice v boxech s důrazem na udržování pořádku na pracovní ploše.

Při plnění úkolů už mohou žáci pracovat samostatně. Je třeba dát čas těm, kteří se na některém kroku zdrží. Naopak není nutné brzdit ty rychlejší. Bylo by dobré, aby všichni žáci zvládli naprogramovat alespoň příkaz ovládání motoru a nahrát program do robotu. Pro připojování kostky k počítači používáme USB kabel (je-li dostupný). Připojením přes Bluetooth není třeba hodinu zatěžovat. Kontrolujte žáky průběžně, diskutujte společně.

2.2.2 Reakce na senzory

Cíl hodiny:

Žák se naučí vytvářet programy, ve kterých robot reaguje na hodnoty přečtené ze senzorů.

Předchozí znalosti:

Žák umí připojit robot k aplikaci a nahrát do něj vytvořený program. Zná příkazy pro ovládání motorů.

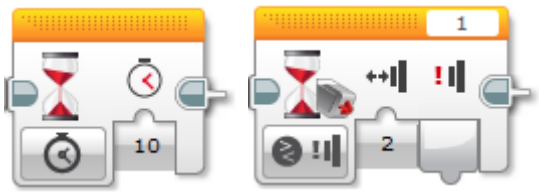
Potřebné pomůcky:

- Programovatelná kostka
- Jeden střední (nebo velký) motor
- Senzor dotyku a senzor barev/světla
- 2–3 kabely na propojení
- USB kabel k propojení počítače a kostky

Úkoly:

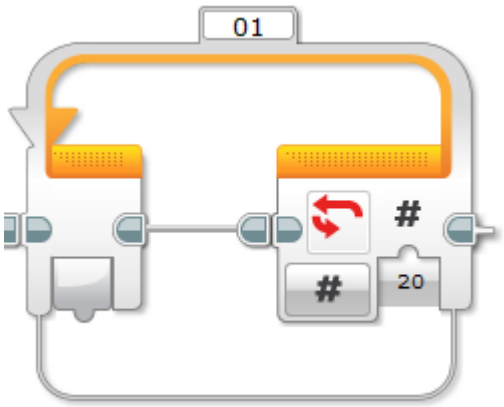
1. Připojte kostku k počítači a propojte ji s aplikací. Dále ke kostce připojte střední motor (port A), senzor dotyku (port 1) a senzor barev/světla (port 3).

2. Vytvořte program, který 10 sekund čeká (viz obrázek 14 vlevo), poté na 5 sekund spustí motor a skončí.



Obrázek 14: Příkazy k čekání po určitý čas (vlevo) či na zadanou hodnotu senzoru (vpravo).

3. Vytvořte program, který 1 sekundu čeká, poté jednou otočí motorem a toto se zopakuje dvacetkrát (viz obrázek 15).



Obrázek 15: Cyklus opakující příkazy uvnitř.

4. Vytvořte program, který spustí motor po kliknutí (viz obrázek 14 vpravo) na tlačítko (dotykový senzor) a po dalším kliknutí motor nechá volně doběhnout (zastaví bez brzdy).

5. Vytvořte program, který při stisknutí tlačítka motor spustí a při jeho uvolnění motor vypne.

6. Program upravte, aby bylo možné motor spouštět opakovaně po libovolně dlouhou dobu.

7. Vytvořte program, který spustí motor při ukázání zelené barvy (přiblížení před senzor) a vypne motor při ukázání červené.

8. Vytvořte program, který okamžitě spustí motor a při zhasnutí (zakrytí senzoru světla) motor vypne.

9. Vytvořené programy si uložte.

Komentář:

Nediktujte žákům postup. Zkuste je nechat na potřebné příkazy přijít samostatně. Příkaz čekání nejspíše žáci zvládnou najít a správně použít. O něco náročnější by mohl být příkaz opakování. Někteří možná budou kopírovat příkazy čekání a spouštění motoru dvacetkrát za sebe. Nechte je to udělat. Až pak jim ukažte, že to lze udělat jednodušeji a rychleji. Dále jim zdůrazněte, že první postup není správně a odteď nadále důsledně vyžadujte použití smyčky. Stejně tak budou možná žáci potřebovat poradit, že příkaz Wait umožňuje čekat na hodnoty senzorů. Na tuto možnost je po chvíli můžete upozornit, nikdy však nezasahujte vlastnoručně do jejich programu. Pouze nabízené možnosti popisujte ústně.

Je dobré nechat žáky pracovat samostatně, nebylo by však dobré zadat úkoly a dále se žákům nevěnovat. Během hodiny mezi žáky procházejte, ptejte se, jak se jim práce daří a zda něco nepotřebují. Kontrolujte splněné úkoly. Někteří žáci mají při nezdaru tendenci od problému odbíhat, bojí se zeptat učitele sami od sebe. Někteří žáci naopak komunikují s učitelem dobře a hodně. Je třeba neustále vyvažovat pozornost mezi všechny žáky.

Při kontrole vyřešených úkolů kontrolujte nejen funkčnost řešení, ale i samotný zdrojový kód. Přestože úlohy jsou v této hodině jednoduché a programy budou krátké, je třeba už od začátku dbát na kulturu kódu. V následujících hodinách se budou programy zvětšovat a špatné návyky hůře odstraňovat.

Je třeba žákům během hodiny (obzvláště na jejím konci) zdůrazňovat, aby si vytvořené programy ukládali. K některým se pravděpodobně budou později ještě vracet.

2.2.3 Kompas

Cíl hodiny:

Žák se seznámí s gyroskopickým senzorem. Naučí se používat příkazy pro čtení senzorů a propojovat hodnoty jednotlivých příkazů.

Předchozí znalosti:

Žák zná příkazy pro čekání a řízení motorů. Umí připojit a naprogramovat kostku.

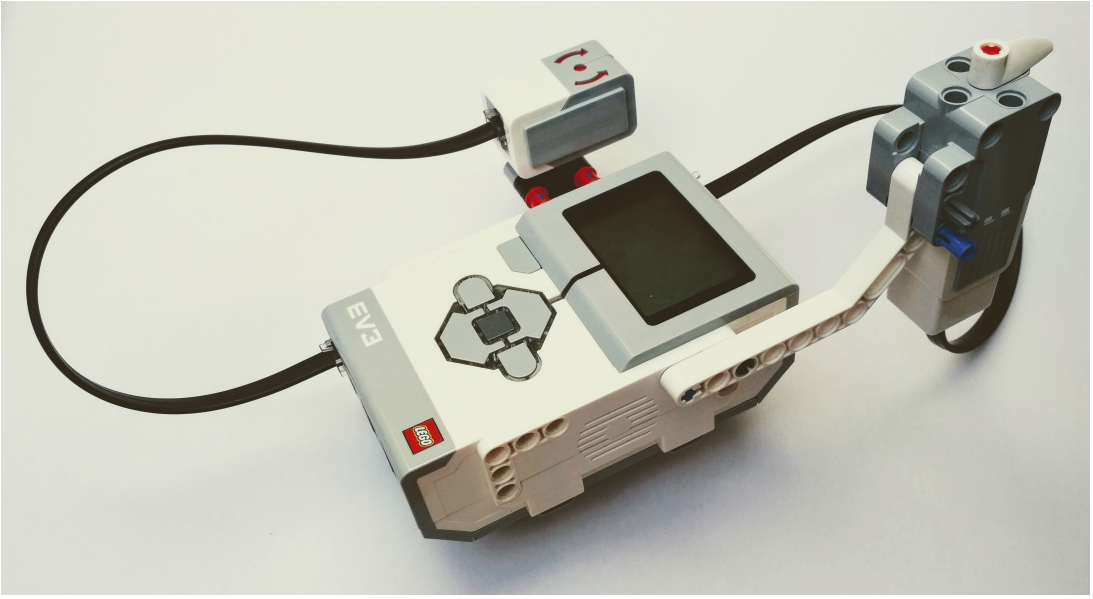
Potřebné pomůcky:

- Programovatelná kostka
- Jeden střední motor a gyroskopický senzor
- Konstrukční dílky a kabely k připojení ke kostce

Úkoly:

1. Spusťte aplikaci a ve vstupním menu vyberte možnost Building Instructions. Dále zvolte Building Ideas a postupně podle návodu připojte ke kostce gyroskopický senzor (Gyro Sensor – Brick).

2. Podobně připojte střední motor (návod Medium Motor – Brick) s jedinou výjimkou: k připojení motoru použijte větší pravoúhlý bílý nosník tak, aby osa otáčení motoru mířila svisle (viz obrázek 16).



Obrázek 16: Ukázka sestaveného kompasu.

3. Připojte kostku k aplikaci pomocí Bluetooth. Kabel by překážel, až budete s kostkou pohybovat/otáčet. Na kostce i počítači spusťte Bluetooth. Na počítači spusťte aplikaci Mindstorms a otevřete nový program. V seznamu zařízení v pravé dolní části vyberte Vaši kostku a připojte ji.

4. Naprogramujte kompas. Pokaždé, když se kostka otočí, otočí se také střelka kompasu umístěná na středním motoru. Přesně o stejný úhel.

5. Ke kostce připevněte tlačítko (dotykový senzor) s možností kompas zkalibrovat. Po jeho stisknutí se střelka začne otáčet. Po uvolnění se střelka zastaví a gyroskopický senzor se resetuje.

Komentář:

Připevnění senzorů a motorů je snadné, takže by ho žáci zvládli nejspíše i bez návodu. Použití návodu bude však důležité v příštích hodinách, proto je dobré, aby se už nyní žáci naučili s návodem pracovat. Zrovna tak připojení kostky

pomocí Bluetooth ještě není zcela nutné, je však příhodné se ho právě v této hodině naučit nastavovat. Úkol s vytvořením kompasu se zaměřuje na příkaz pro čtení hodnot senzoru a napojení hodnot pro využití v dalších příkazech.

Vytvořený kompas bude pravděpodobně velmi nepřesný. Střelka možná nebude odpovídat v reálném čase. Není třeba dbát na tyto detaily. Podstatné je, aby žáci správně použili příkazy čtení hodnost gyroskopického senzoru a propojení výstupních parametrů jednoho příkazu na vstup dalšího příkazu. Přesnost chování mohou zlepšit matematické příkazy. Pokud budeme počítat s celkovým otočením od začátku, nebude docházet ke sčítání chyb. Tuto možnost můžete poradit nadanějším žákům, kteří budou s úkolem rychleji hotovi nebo ho budou chtít mít lepší. Není však nutné matematickými příkazy zatěžovat všechny žáky.

2.2.4 Sestavení základního robotu

Cíl hodiny:

Žák sestaví základní pojízdný robot podle návodu a naučí se používat příkazy pro jeho pohyb (jízda přímo i do zatáčky) pomocí dvou motorů.

Předchozí znalosti:

Žák již z minulých hodin disponuje základní orientací v prostředí kostky a aplikace. Zná také příkaz ovládání motoru, na který navazují příkazy ovládání robotu pomocí dvou motorů.

Potřebné pomůcky:

- Programovatelná kostka
- Dva velké motory (a kabely k propojení s kostkou)
- Konstrukční díly k sestavení robotu (viz návod)

Úkoly:

1. Spusťte aplikaci a v levém menu vyberte čtvrtou možnost (Building Instructions). V podmenu zvolte první možnost (Building Ideas) a model Driving Base.
2. Sestavte robot podle návodu.
3. Pomocí Bluetooth připojte robot a naučte (naprogramujte) ho, aby popojel o kousek dopředu a pak se vrátil zpět.
4. Naprogramujte robot, aby se otočil na místě jednou kolem dokola.
5. Naprogramujte robot, aby objel kružnici o poloměru přibližně 20 cm.

6. K pohybu můžete využívat příkazy Move Tank a Move Steering. Zkuste udělat každý z předchozích programů oběma způsoby (využitím jednoho i druhého typu příkazů).

Komentář:

Smyslem této hodiny jsou dvě hlavní věci. První je sestavení robotu. Robot nemusí být po skončení hodiny rozebírán a může být využit v následujících hodinách, což ušetří spoustu času, který by jinak zabralo opětovné skládání robotu. Zkušený stavitel zvládne robot sestavit i za 15 min. Vzhledem k tomu, že většina žáků staví takový složitější robot nejspíš poprvé, počítejme s tím že jim sestavení může trvat 45 minut nebo více. Žáci budou déle hledat jednotlivé dílky, některé kroky budou muset opakovat či opravovat.

Druhým smyslem hodiny je, aby žáci pochopili řízení pohybu pomocí dvou motorů. Když jeden motor jede rychleji a druhý pomaleji (případně vůbec nebo opačným směrem), robot zatáčí. Žáci nemusí stihnout všechny úkoly, řízení robotu bude dostatečně procvičeno i v dalších hodinách. Mohou naopak přidat vlastní pokusy v případě nadbytku času.

2.2.5 Řízení robotu

Cíl hodiny:

Žák si procvičí řízení robotu na složitější dráze.

Předchozí znalosti:

Žák má sestavený robot, umí ho programovat a zná příkazy k řízení robotu pomocí dvou motorů.

Potřebné pomůcky:

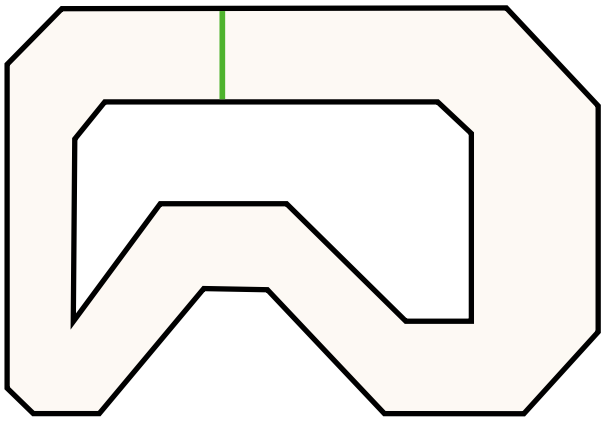
- Základní robot (Driving Base)
- Vyznačená dráha (podle možností, postačí barevná lepicí páska)

Úkoly:

1. Naprogramujte robot tak, aby projel danou dráhu.
2. Zdokonalujte program, aby robot dráhu projel co nejrychleji a co nejpřesněji.

Komentář:

Dráhu, kterou má robot projet, je dobré připravit předem. Je možné ji například vyznačit lepicí páskou na zemi. Dráha tak může být vhodně velká a robot nemá kam spadnout. Pokuste se udělat dráhu složitější: různě dlouhé úseky, různě ostré zatáčky, různě úzká místa (vždy však tak, aby robot mohl projet). Příklad takové dráhy ukazuje obrázek 17.



Obrázek 17: Příklad dráhy (1:20) přibližně odpovídající vzorovému programu.

Můžete pak uspořádat soutěž/závod, který robot projede dráhu nejrychleji, přičemž za každé přejetí okrajové čáry dostane soutěžící robot několik vteřin navíc. Při samotném projíždění dráhy pak velmi citlivě záleží na výchozím umístění robotu. Řízení motorů navíc není úplně přesné, robot tedy nejede pokaždé zcela stejně. Penalizace je doporučeno stupňovat (např.: 1. kolo vjede na čáru, 2. kolo vyjede mimo čáru, 3. robot úplně opustí dráhu nebo není schopen se vrátit).

Není možné naprogramovat projetí celé dráhy najednou a napoprvé, žáci budou potřebovat mnohokrát program vyzkoušet. V této hodině není důvod jim toto zakazovat. Do soutěže můžete započítat až poslední pokus na konci hodiny. Nebo můžete do výsledku soutěže započítat počet potřebných pokusů k finálnímu naprogramování každého programu.

Na konci závodu lze žáky odměnit za různé aspekty programu. Například za nejrychlejší či nejpřesnější projetí dráhy nebo za originální řešení. Je vhodné najít u každého žáka něco, za co jej lze odměnit či pochválit. Ideálně by měl být odměněn každý žák, aby nedocházelo k posílení negativní motivace u těch méně úspěšných.

Příkazy použité v programu jsou jednoduché a žáci je znají z předchozích hodin. Komplikovanější bude nastavení parametrů daných příkazů, aby robot projel dráhu přesně obzvláště v náročných úsecích trati. Ještě náročnější bude citlivě odhadnout kompromis mezi rychlostí a přesností jízdy.

2.2.6 Ultrazvukový senzor

Cíl hodiny:

Cílem hodiny je vyzkoušet a pochopit práci s ultrazvukovým senzorem. Žák se krátce seznámí s podmíněným příkazem (Switch).

Předchozí znalosti:

Příkazy ovládání robotu, čekání na hodnotu senzoru, nekonečná smyčka.

Potřebné pomůcky:

- Základní robot (Driving Base)
- Ultrazvukový senzor
- Příslušné konstrukční díly a kabel k připojení senzoru

Úkoly:

1. Najděte v aplikaci návod k připojení ultrazvukového senzoru (Building Instructions – Building Ideas – Ultrasonic Sensor – Driving Base).
2. Podle návodu připojte k základnímu robotu ultrazvukový senzor.
3. Naprogramujte robot tak, že pojede vpřed, dokud se 20 cm před ním neobjeví překážka. Poté program skončí.
4. Upravte předchozí program: Pokaždé, když jede dopředu, kontroluje překážky. Pokud je překážka blíže než 10 cm, robot se zastaví. Jakmile je překážka odstraněna, robot se opět rozjede vpřed.
5. Naprogramujte robot tak, že při zjištění překážky se zastaví, čeká pět sekund a poté, pokud je překážka odstraněna, pokračuje v cestě, jinak program skončí.
6. Upravte předchozí program: pokud není do pěti sekund překážka odstraněna, robot se otočí a jede zpátky, odkud přijel. Cestou zase kontroluje vzdálenost předmětů před ním. Při rozhodnutí, zda se má otočit, či pokračovat, použijte podmínku (příkaz Switch).
7. S použitím ultrazvukového senzoru a vhodně umístěné překážky určete, kolik stupňů otočení motoru je zapotřebí, aby robot ujel 50 cm.

Komentář:

V některých případech možná budou žáci potřebovat poradit. Nechte je však chvíli, ať nejprve zkusí na řešení přijít sami. Až po nějaké době jim můžete pomoci. Příkaz Switch bude pravděpodobně kontrolovat stejnou podmínku, na kterou čeká předchozí příkaz kontrolující překážku. Při otočení zkontrolujte, aby se robot otočil co nejpřesněji čelem vzad. Pokud zbude čas, můžete otočení ověřit pomocí gyroskopu. U posledního úkolu ověřte, že je ke změření vzdálenosti opravdu správně použit ultrazvukový senzor a ke zjištění úhlu senzor otočení motoru. Ověřte také, že byl před měřením senzor otočení vynulován.

Ke konci hodiny diskutujte se žáky fungování příkazu Switch. Pokládejte otázky typu: „Co by se stalo, když…?“ Ověřte správné pochopení jeho funkce. Pro větší názornost můžete využít grafické znázornění (např. na tabuli) větví programu závislých na splnění podmínky.

2.2.7 Manipulace s objektem

Cíl hodiny:

Žák si vyzkouší programovat robot tak, aby interagoval s okolím. Dále se naučí propojovat hodnoty mezi různými bloky programu.

Předchozí znalosti:

Žák už z předchozí hodiny ovládá práci s ultrazvukovým senzorem. Umí ovládat motory a číst data ze senzorů.

Potřebné pomůcky:

- Základní robot (Driving Base)
- Gyroskopický a ultrazvukový senzor
- Střední motor
- Kabely k připojení motoru a senzorů
- Součástky ke stavbě manipulačního ramene a kvádru

Úkoly:

1. Podle návodu v aplikaci sestavte a připojte k robotu manipulační rameno (Building Instructions – Building Ideas – Medium Motor – Driving Base).
2. Podle návodu v aplikaci přidejte k robotu ultrazvukový senzor (Building Instructions – Building Ideas – Ultrasonic Sensor – Driving Base).
3. Podle návodu v aplikaci postavte kvádr, se kterým bude robot manipulovat (Building Instructions – Building Ideas – Cuboid).

4. Naprogramujte robot tak, že přijede (podle ultrazvukového senzoru) těsně ke kvádru, poté ho uchopí a vrátí se s ním (pozadu) na původní místo, ze kterého vyjel.

5. Připojte k robotu gyroskopický senzor (port 2). Tentokrát není třeba používat návod. Senzor připojte ke kostce jednoduše dvěma spojovacími dílky (viz obrázek 18).

Obrázek 18: Příklad připojení gyroskopického senzoru.

6. Upravte předchozí program tak, aby robot odvezl kvádr na výchozí místo popředu (tzn. po uchopení objektu se otočí o 180 °). Přesnost otočení zajistěte pomocí gyroskopického senzoru.

Komentář:

Zkontrolujte žáky, jestli se robot přibližuje ke kvádru opravdu pomocí ultrazvukového senzoru a zda se otáčí opravdu s pomocí gyroskopu. V této hodině je více konstrukčních kroků a méně programování. Přesto je v programování jedna velmi důležitá část, a to návrat robotu na původní místo, kde je třeba použít data ze senzoru otočení v jednom z velkých motorů. Čtvrtý úkol bude jednodušší. Pokud byl senzor otočení na začátku programu resetován, bude stačit při návratu čekat, než bude úhel otočení menší nebo roven nule.

V šesté úloze však již předchozí postup není možný, neboť obě cesty podniká robot směrem vpřed. Žáci by si na tomto kroku měli procvičit použití hodnoty jednoho bloku (přečtení úhlu otočení motoru) v bloku dalším. Danou úlohu lze řešit i jinak, bylo by však dobré, aby žáci o popsané možnosti alespoň věděli.

Gyroskopický senzor je připojen jednoduše ke kostce, neboť připevnění dle návodu je vratké a ne zcela stabilní. Kvůli množství konstrukčních úkolů je možné, že budou žákům chybět některé konstrukční dílky. Obzvláště v mírně neúplných sadách. Toto není ničemu na škodu. Žáci si musí s problémem poradit. Využít jiné dostupné dílky k vyřešení úkolů.

2.2.8 Senzor barev/světla

Cíl hodiny:

Žák dokáže použít senzor barev/světla k detekci čáry vyznačené na podložce.

Předchozí znalosti:

Žák by již měl zvládat použít nekonečnou smyčku (cyklus), příkazy k řízení robotu a čekání programu na hodnoty senzorů.

Potřebné pomůcky:

- Základní robot (Driving Base)
- Senzor barev/světla
- Konstrukční dílky a kabel k připojení
- Vyznačená čára na zemi (dle možností postačí barevná lepicí páska)

Úkoly:

1. Spusťte aplikaci a podle návodu k robotu připojte senzor barev/světla (Building Instructions – Building Ideas – Color Sensor Down).
2. Naprogramujte robot tak, že se po spuštění programu rozjede vpřed. Pokud robot najede na černou čáru, zastaví.
3. Upravte následovně předchozí program. Jakmile robot najede na čáru, zastaví a 5 sekund čeká, poté se otočí (pro přesnost otočení můžete použít gyroskopický senzor) a vrátí se do výchozí pozice.
4. Naprogramujte robot tak, že pokud ho postavíte na čáru a spustíte program, robot čáru sleduje.
5. Pokud gyroskopický senzor ukazuje otočení o více než 360 °, může to znamenat dvě věci. Buďto už robot objel celou dráhu, nebo z dráhy vyjel a točí se v kruhu. Zajistěte, aby v takovém případě program skončil.

Komentář:

Čáru můžete použít libovolnou. Je možné třeba na zem nalepit barevnou lepicí pásku. Čára by však měla mít tloušťku alespoň 2 cm, aby ji barevný senzor byl schopen rozeznat a včas na ni zareagovat. Dále je potřeba, aby podklad byl pokud možno jednobarevný a co nejvíce kontrastní s barvou lepicí pásky. Může stačit např. světle šedé nebo béžové linoleum. Při tvorbě této práce nebyla podlaha s vhodným povrchem dostupná, proto byla využita deska z překližky o rozměrech přibližně $1,1 \times 1,6$ m natřená bílou barvou.

Ve čtvrtém úkolu (sledování čáry) diskutujte s žáky princip fungování programu hromadně. Někteří na něj nejspíš nepřijdou hned sami. Vytvořte dostatečně různorodou čáru, aby měla ostré zatáčky i dlouhé rovné úseky. Můžete pak spolu se žáky sledovat, jaký vztah má rychlost projetí čáry a přesnost v zatáčkách (nevyjetí z dráhy). Upravujte různě parametry motorů, hledejte optimální nastavení. Ptejte se žáků, jak jsou spokojeni s výsledky svého programu. Případně diskutujte možnosti, jak program dále vylepšit, aby výsledek lépe odpovídal žákově představě.

2.2.9 Světlo a zvuk kostky

Cíl hodiny:

Žák dokáže použít světelnou signalizaci na kostce (pod tlačítky) a ovládat reproduktor kostky.

Předchozí znalosti:

Žák zvládá používat příkazy pro řízení robotu, čekání na událost a nekonečnou smyčku. Setkal se už také s větvením pomocí příkazu Switch. Dále umí pracovat s daty vzdálenosti z ultrazvukového senzoru.

Potřebné pomůcky:

- Základní robot (Driving Base)
- Ultrazvukový senzor
- Kabel a konstrukční díly k připojení

Úkoly:

1. Naprogramujte robot, aby svítil jako semafor:
    - Zelené světlo bude svítit 5 sekund, oranžové 2 sekundy a červené 5 sekund. Pak opět 2 sekundy oranžové a znovu od zeleného.
    - Toto se zopakuje třikrát a poté se semafor vypne. Bude (neomezeně dlouho) pouze blikat oranžové světlo.
    - Při rozsvícení oranžového světla pro přechod ze zelené na červenou se ozve výstražný zvuk. Na červené světlo robot řekne „Stop!“. Ve vypnutém stavu (blikání oranžového světla) se žádné zvuky neozývají.
2. Nyní použijeme ultrazvukový senzor k naprogramování hlídkujícího robotu:
    - Robot jede vpřed, dokud se nepřiblíží k překážce (na 5 cm). Poté zastaví a 5 sekund bliká oranžové světlo.

- Pokud je po 5 sekundách překážka odstraněna, rozsvítí se zelené světlo a robot pokračuje vpřed. Jinak se spustí poplach. Začne blikat červené světlo a ozve se výstražný zvuk. Poplach trvá do vypnutí programu.
- Robot nespustí poplach u stěny místnosti, ale dříve, než k ní dojede, se otočí a pokračuje zpět k výchozí pozici. Na ní se opět otočí a pojede stejným směrem jako na začátku.

Komentář:

V prvním úkolu jde zejména o naučení a procvičení příkazů k ovládání signálního světla a zvuku kostky. Při kontrole úkolů můžete žákům ukázat, kde se ve vlastnostech projektu ukládají použité zvuky. Pravděpodobně zde budou uloženy všechny zvuky, které žáci při programování zkoušeli, i když některé z nich nakonec v hotovém programu nejsou vůbec použity. Takové zvuky je dobré ze seznamu vymazat, aby soubor projektu nebyl větší než je nutné. Kontrolujte, aby oranžové světlo ve stavu vypnutého semaforu blikalo opravdu neomezeně dlouho, nikoli pouze na dlouhou dobu.

Pokud bude více žáků současně zkoušet své programy, může v učebně vzniknout větší množství hluku. Požadujte po žácích ztišení zvuků například na desetinu. Někteří se možná budou pokoušet ztišit zvuk v nastavení kostky. Tam však lze ztišit pouze systémové zvuky. Hlasitost zvuku je třeba snížit úpravou parametru v programu.

Ve druhém úkolu se možná budou žáci marně snažit přijít na to, jak pomocí ultrazvukového (nebo jiného) senzoru rozlišit překážku a konec místnosti, ale není to potřeba. Místnost je stále stejně velká a můžeme předpokládat stabilní výchozí pozici robotu. Stačí tedy kontrolovat pouze maximální ujetou vzdálenost (otočení motorů). Určitě pomůže také příkaz pro reset senzoru otočení v motorech.

2.2.10 Tlačítka a displej kostky

Cíl hodiny:

Žák dokáže v programu číst tlačítka kostky a vypisovat informace na displej. Dále se naučí použít příkaz Switch s více než dvěma možnostmi.

Předchozí znalosti:

Žák zvládá používat příkazy pro řízení robotu, čekání na událost a nekonečnou smyčku. Setkal se s větvením pomocí příkazu Switch, jehož použití nyní rozšíří.

Potřebné pomůcky:

- Základní robot (Driving Base)

Úkoly:

1. Naprogramujte robot, který bude možné řídit pomocí tlačítek a displeje na kostce. Program pracuje ve dvou fázích: čekání na příkaz a vykonávání příkazu. Ve fázi čekání na příkaz se na displeji kostky objeví nápis: „Čekám, zadejte příkaz.“ Po kliknutí na libovolné tlačítko se na displeji objeví symbol příkazu, robot počká půl sekundy a vykoná zadaný příkaz.

   - Vpřed – robot popojede vpřed.
   - Vzad – robot popojede vzad.
   - Vpravo – robot se na místě otočí doprava.
   - Vlevo – robot se na místě otočí doleva.
   - Pozdrav (prostřední tlačítko) – na displeji se zobrazí oči a robot uživatele pozdraví: „Hello!“

2. Upravte program tak, aby se po vykonání příkazu robot vrátil do první fáze čekání na stisk tlačítka. Program tak pokračuje stále dokola, dokud jej neukončíme.

Komentář:

Na začátku hodiny je dobré připomenout žákům funkci podmíněného příkazu Switch a poradit jim, jeho použití k rozlišení tlačítek. Možnost přidání dalších případů do příkazu Switch nemusíme žákům prozrazovat hned, ale až po nějaké době. Pokud na ni přijdou sami, tím lépe. Pokud na ni nepřijdou, je dobré, když problém vyřeší jinak a poté uvidí, že zmíněná možnost přinese značné zjednodušení.

Čekání půl sekundy po zadání příkazu má ten význam, aby člověk stihl uvolnit prst od tlačítek, než se robot začne hýbat. Míra otočení doprava či doleva není stanovena. Ideálně by se měl robot otočit o 90 °, ale není to nutné, úhel otočení může být menší.

2.2.11 Zpracování textu

Cíl hodiny:

Žák dokáže spojovat různé (číselné i textové) hodnoty do delších textů a vypisovat je na displej.

Předchozí znalosti:

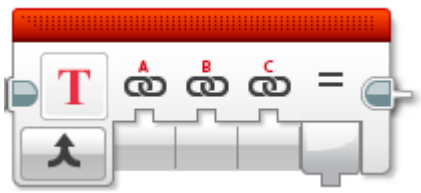
Žák umí používat senzory a číst jejich hodnoty. Zvládne vypsat text na displej kostky.

Potřebné pomůcky:

- Základní robot (Driving Base)
- Ultrazvukový a barevný senzor
- Kabely a konstrukční díly k připojení

Úkoly:

1. Podle návodu připojte k robotu ultrazvukový senzor a senzor barev (mířící vpřed).
2. Robot se v tomto úkolu nebude nijak pohybovat, pouze při spuštění vypíše na první řádek displeje měřenou vzdálenost a na druhý řádek číslo měřené barvy. Poté počká 10 sekund a skončí.
3. Měřené hodnoty se budou vypisovat v reálném čase. V intervalech jedné desetiny sekundy.
4. Upravte výpis vzdálenosti a barvy:
   - Na prvním řádku bude text „Vzdálenost: X cm.“, kde X bude vzdálenost naměřená ultrazvukovým senzorem zaokrouhlená na celé centimetry. Bude se vám hodit příkaz Text – Merge (viz obrázek 19).



Obrázek 19: Příkaz ke spojování textových řetězců.

   - Na druhém řádku bude text „Barva: Y“, kde Y bude název barvy, tedy „červená“, „zelená“, „modrá“...
5. Vyzkoušejte funkčnost výpisu hodnot na co největším počtu pokusů.

Komentář:

Tato hodina by měla být relativně snadná, žáci už umí zpracovat data ze senzorů, umí vypisovat text na displej i propojovat v programu hodnoty výstupních a vstupních parametrů. Jediným novým příkazem je spojení více textů do jednoho, případně zaokrouhlení vzdálenosti. Smyslem této hodiny je ulehčit hodinám příštím, ve kterých je složitější výpis na displej součástí komplexnějšího zadání.

Výpis měřené vzdálenosti bude snadný. O něco náročnější bude výpis barvy, která je měřena jako čísla 0–7, ale vypsána má být jako jméno barvy. Někteří žáci možná budou potřebovat poradit použití větvení (Switch) s více případy.

Výpis hodnot v reálném čase by se měl obnovovat v intervalech jedné desetiny sekundy. To je přibližná rychlost, v jaké je lidské oko schopno zaznamenat změnu. Senzory jsou sice schopné číst tisíc hodnot za sekundu, ale pokud nastavíme rychlou smyčku bez zpomalovacího příkazu Wait, displej nebude stíhat obnovovat hodnoty. To se projeví oscilací a špatnou viditelností vypsaného textu.

2.2.12 Proměnné

Cíl hodiny:

Žák se naučí používat proměnné. Dokáže vytvořit program, který dělá víc věcí najednou.

Předchozí znalosti:

Žák zvládl minulou lekci s použitím senzoru barev/světla a umí použít ultrazvukový senzor vzdálenosti.

Potřebné pomůcky:

- Základní robot (Driving Base)
- Senzor barev/světla a ultrazvukový senzor
- Konstrukční dílky a kabely k připojení
- Vyznačená čára na zemi (dle možností postačí barevná lepicí páska, bude potřeba mít čáry více barev)

Úkoly:

1. Podle návodu připojte k robotu senzor barvy/světla a ultrazvukový senzor.
2. Použijeme program z dřívější hodiny. Robot bude opět sledovat čáru na zemi. S použitím proměnné (Variable) si však robot zapamatuje barvu přečtenou při spuštění programu. Tuto barvu pokládá za barvu čáry. Může tedy sledovat čáru libovolné barvy, pouze na ní musí stát při spuštění programu.
3. K programu přidejte detekci překážky. Robot sleduje čáru, dokud je před ním volno. Než by narazil na překážku, zastaví. Po odstranění překážky robot pokračuje po čáře dál.
4. Počet překážek bude robot počítat v další proměnné a zobrazovat jej na displeji.

Komentář:

Tato hodina je náročná hlavně na pochopení principu fungování proměnných. Někteří žáci pochopí rychle a stihnou všechny úlohy. Jiní se někde zastaví. Nevadí, že nestihnou všechny úkoly. Důležitější je, aby pochopili základy práce s proměnnými. Procvičení se jim dostane i v dalších hodinách. Ke konci hodiny zopakujte s žáky tematiku proměnných. Žák by měl vědět, co je to proměnná a k čemu slouží. Zdůrazněte žákům, že proměnné jsou v programování opravdu důležité. Také v dalších hodinách je budou žáci pravidelně využívat.

Při kontrole úkolů není třeba dbát tolik na bezproblémovou funkčnost programu. Menší nedostatky je možné prominout. Naopak dbejte na správnost použití proměnných a ptejte se žáků na jednotlivé části jejich kódu: „Proč jsou příkazy použity právě takto?“, „Co tato část znamená nebo jak funguje?“ a podobně.

Poznámka ke vzorovému kódu: U kódů úkolů 3 a 4 jsou uvedeny dvě možnosti. První je spolehlivější, druhá však lépe využívá proměnných, proto je zde uvedena také. Všimněte si rozdílné hodnoty vzdálenosti při detekci překážky a detekci jejího odstranění kvůli eliminaci zákmitů při mezních hodnotách. Při zákmitech by se robot mohl chovat nepředvídatelně nebo v posledním úkolu napočítat více překážek. Při současné detekci překážky a změně detekované barvy je možné, že robot bude pokračovat dál proti překážce. Problém je možné odstranit použitím podmínky a logické proměnné, avšak za cenu mnohem složitějšího kódu.

2.2.13 Proměnné podruhé

Cíl hodiny:

Žák si procvičí použití proměnných a práci s tlačítky a displejem kostky.

Předchozí znalosti:

Žák umí v programu použít proměnnou, přiřadit jí hodnotu nebo hodnotu přečíst. Zná příkazy pro řízení robotu, čtení tlačítek, výpis na displej, větvení a další.

Potřebné pomůcky:

- Základní robot (Driving Base)

Úkoly:

1. Vytvořte nový program, který bude podobně jako program v některé z minulých hodin sloužit k ovládání pohybu robotu pomocí tlačítek na kostce. Po spuštění programu je robot v režimu zadávání příkazu. Na rozdíl od předchozího programu však budeme různými tlačítky ovládat různé parametry jednoho příkazu (Move – Steering).

- Tlačítky doprava a doleva se mění míra zatočení. Jedno stisknutí tlačítka změní hodnotu zatočení o 5 %.
- Tlačítky nahoru a dolů se mění ujetá vzdálenost v počtu otoček. Jedno stisknutí tlačítka změní počet otoček motoru o 1.
- Prostředním tlačítkem se spustí vykonání příkazu. Robot počká půl sekundy, vykoná příkaz se zadanými parametry a vrátí se do režimu zadávání příkazu.

2. Upravte program tak, aby volené parametry zobrazil na displeji kostky.
   - Čísla parametrů zobrazujte vždy kladná.
   - Za číslem vypište ještě směr, tedy u zatočení vpravo, či vlevo a u jízdy vpřed, nebo vzad.
   - Míra zatočení se udává v procentech, maximální hodnota je tedy 100 %. Zajistěte, aby danou hodnotu nebylo možné přesáhnout.

Komentář:

Zadané úkoly by mohli žáci zvládnout sami. Potřebné znalosti k tomu už mají. V podstatě se jedná pouze o kombinaci prvků několika předchozích programů. Podstatou této hodiny je, aby se žáci učili použít nástroje, které se dříve naučili na jednoduchých příkladech, k vytvoření komplexnějšího programu. Klíčový krok postupu je uvědomění si, že každý parametr bude uložen zvlášť ve své proměnné.

Dalším podstatným prvkem je patřičná úprava hodnot před výpisem na displej. Číslo musí být načteno z proměnné. Dále musí být určeno jeho znaménko a podle něj vybrán patřičný směr. Před samotným výpisem čísla je třeba ještě získat jeho absolutní hodnotu. Nejjednodušší je použít přímo matematický příkaz absolutní hodnoty, ale je možné problém řešit i jinak. Pokud někteří žáci budou mít jiná řešení, na možnost příkazu pro absolutní hodnotu je upozorněte. Pochvalte je však, že samostatně a správně problém vyřešili, což je mnohem důležitější, než použití okrajového příkazu pro absolutní hodnotu.

2.2.14 Ovládání pomocí mobilu

Cíl hodiny:

Žák se naučí používat aplikaci Mindstorms Commander pro zařízení se systémem Android nebo iOS a s její pomocí ovládat robot.

Předchozí znalosti:

Žák umí použít příkazy pro řízení robotu, chápe princip řízení pomocí dvou motorů a zná funkci senzorů a práci s nimi.

Potřebné pomůcky:

- Základní robot (Driving Base)
- Střední motor
- Konstrukční díly manipulačního ramene
- Chytrý telefon se systémem Android nebo iOS
- Drobné objekty k manipulaci

Úkoly:

1. Podle návodu připojte k robotu manipulační rameno poháněné středním motorem.
2. Ve svém mobilním zařízení jděte do obchodu s aplikacemi a nainstalujte si aplikaci Mindstorms Commander.
3. Propojte aplikaci s robotem. V aplikaci zvolte možnost Custom robot.
4. Vytvořte ovládací prostředí podle obrázku 20. Použijte následující ovládací prvky:
   - Joystick – pro řízení pohybu robotu
   - Horizontal slider – pro ovládání manipulačního ramene

Obrázek 20: Prostředí k ovládání základního robotu pomocí mobilu.

5. Správnou funkci ovládání vyzkoušejte.

Komentář:

Tato hodina by měla být pro žáky jednodušší a měla by mít spíše oddychový charakter. Je schválně zařazena jako odlehčení mezi náročnější hodiny s proměnnými a hodiny s tvorbou vlastních bloků, které budou následovat. Ovládání robotu pomocí mobilního zařízení má nižší význam pro náš cíl, jímž je výuka základů algoritmizace. Žáky naopak tato hodina nejspíš bude více bavit pro její jednoduchost a hravost.

K vyzkoušení správné funkce ovládání robotu můžete uspořádat například závod v projetí určité dráhy zároveň s manipulací různými předměty. Robot začíná ve výchozí pozici. Na jiném místě je umístěn předmět, ke kterému musí robot přijet a přemístit jej na určené místo. Takto může přemístit několik předmětů. Předměty mohou být umístěny v prostoru nebo u stěny. Předmět umístěný v prostoru může robot pouze odtlačit. Předmět umístěný u stěny odtlačit nelze, k jeho uchopení tedy musí soutěžící použít robotické rameno ovládané středním motorem.

Odpočinkový charakter hodiny nabízí větší možnosti individuální výuky. Žáci se mohou v případě zájmu věnovat vlastním pokusům. Pokud někteří žáci na předchozích hodinách chyběli, mohou v této hodině dohnat, co nestihli. Nebylo by však dobré připravit je o zábavu z ovládání robotu mobilem. Pro náš cíl výuky není sice tato možnost příliš důležitá, ale pro žáky by mohla. Pokud by jim tato možnost byla odepřena, mohla by se jim výuka znechutit.

2.2.15 Tvorba vlastních bloků

Cíl hodiny:

Žák se naučí ze skupiny bloků vytvořit vlastní blok pro zpřehlednění programu.

Předchozí znalosti:

Žák používá příkazy pro řízení robotu a čtení gyroskopického senzoru. Zná příkaz smyčky. Dále žák využije program vytvořený v předminulé hodině (2.2.13).

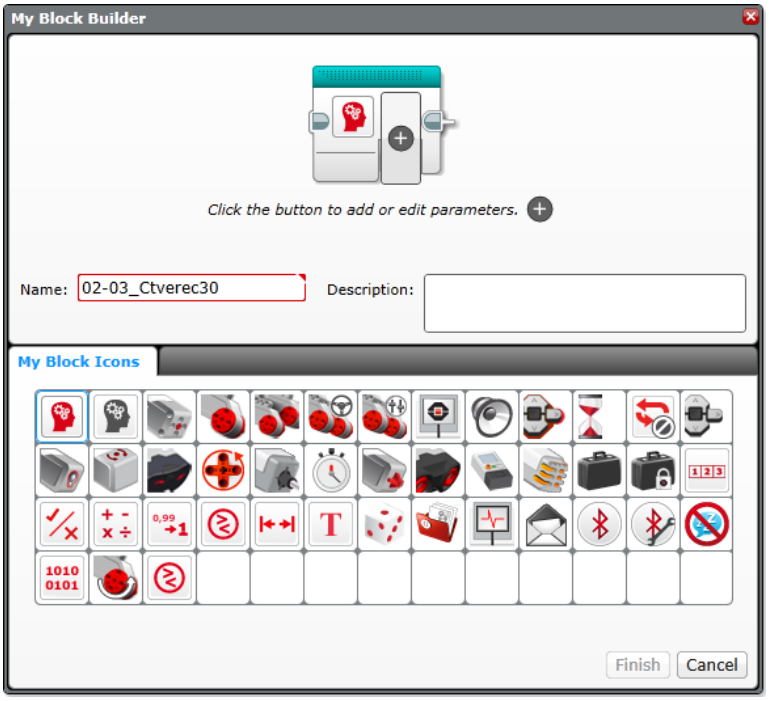
Potřebné pomůcky:

- Základní robot (Driving Base)
- Gyroskopický senzor

Úkoly:

1. Vytvořte program, ve kterém robot objede čtverec o straně 30 cm. Přesnost zatočení o 90 ° zajistěte použitím gyroskopického senzoru.
2. Vyberte všechny bloky programu a volbou možnosti Tools – My Block Builder vytvořte nový blok.

- V nově otevřeném okně (viz obrázek 21) vyplňte název bloku a vyberte novému bloku ikonu.



Obrázek 21: Okno My Block Builder pro volby vlastností nového vlastního bloku.

- Svůj nový blok najdete v dolní nabídce My Blocks (světle modrá). A pomocí záložek (nahoře) můžete přepínat mezi celým programem a úpravou daného bloku. Záložku s vlastním blokem otevřete dvojklikem na daný blok. V levém horním rohu vlastního bloku se nachází tlačítko nastavení, s jehož pomocí můžete znovu otevřít okno vlastností bloku. Blok můžete také najít po otevření záložky Project Properties (nahoře vlevo) v seznamu vlastních bloků.

3. S použitím smyčky a nového bloku vytvořte program, ve kterém robot objede čtverec desetkrát.

4. Otevřete program z dřívější hodiny (2.2.13) pro ovládání robotu tlačítky s použitím proměnných a zjednodušte ho vytvořením a použitím několika vlastních bloků (např. výpis počtu otoček motoru na displej, výpis zatočení na displej, zadávání příkazu).

Komentář:

Jedná se o zcela novou věc a zásadní zlom v programování. Od této hodiny žák může spojovat více bloků do jednoho a vytvářet tak rozsáhlé, avšak přehledné programy. Nemluvě o dalších možnostech jako rekurze apod. Je třeba, aby žák

pochopil, že na místě nového bloku se pouze jednoduše vykonají všechny bloky v něm obsažené. V následující hodině se na této myšlence bude dále stavět.

Jedním z hlavních přínosů vlastních bloků je lepší přehlednost programu. Aby však nový blok této přehlednosti přispěl, je třeba, aby byl správně pojmenován a byla mu vybrána pokud možno odpovídající proměnná. Při kontrole úkolů na tyto zásady dbejte a po žácích je vyžadujte. Jména typu „blok 1“ nebo „ctv“ nejsou vhodná. Místo toho je lepší použít například název „ctverec 30“, ze kterého je možné lépe odhadnout, že robot objede čtverec o straně 30 cm.

Volba ikonky je neméně důležitá. Obecná ikonka pro vlastní blok (hlava s ozubenými koly) by se měla používat co nejméně. Vhodnou ikonu připomínající čtverec sice nemáme. Je však možné použít některou ikonu dvou motorů, jelikož příkaz pro objetí čtverce ovládá motory robotu. Podobně ve volbě názvu a ikony postupujte i u ostatních bloků v dalších úlohách.

2.2.16 Tvorba bloků s parametry

Cíl hodiny:

Žák se naučí tvořit bloky se vstupními parametry.

Předchozí znalosti:

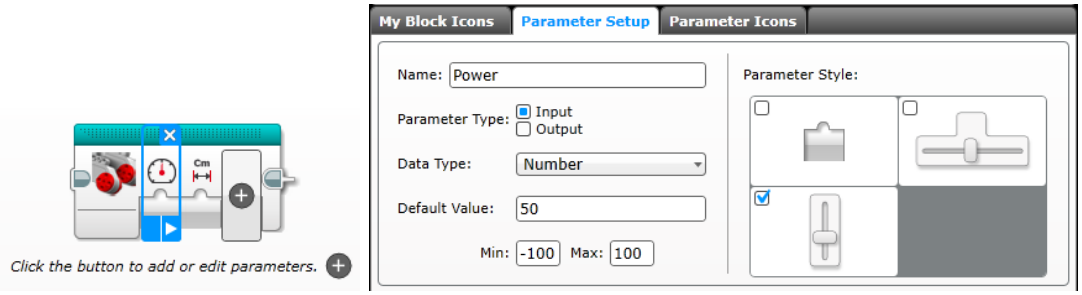
Žák zná příkazy k ovládání robotu a umí vytvářet jednoduché bloky ze sekvence příkazů.

Potřebné pomůcky:

- Základní robot (Driving Base)
- Gyroskopický a ultrazvukový senzor

Úkoly:

1. Stejně jako výchozí bloky Mindstorms i nově vytvářené bloky mohou mít parametry. Tuto možnost nabízí okno s vlastnostmi nového bloku (viz obrázek 22).



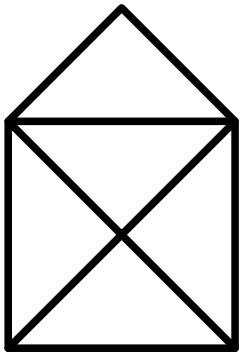
Obrázek 22: Části okna My Block Builder pro volbu vlastností parametrů bloku.

2. S použitím ultrazvukového senzoru určete převodní vzorec mezi ujetými centimetry a stupni otočení motorů (např. o kolik stupňů se musí motory otočit, aby robot ujel 100 cm).

3. Po zjištění tohoto vztahu vytvořte nový blok „Jeď vzdálenost“, který bude mít dva číselné parametry: sílu motorů (rychlost) a ujetou vzdálenost v cm.

4. Stejně postupujte pro otočení robotu (na místě) o určitý úhel. Ke zjištění vztahu použijte gyroskopický senzor a vytvořte příkaz „Otoč se“ se dvěma parametry: silou motorů a úhlem otočení robotu.

5. S použitím pouze těchto vlastních bloků vytvořte program, ve kterém robot objede tvar domku (viz obrázek 23). Zkuste si nejprve nakreslit na papír domek z obrázku jedním tahem.



Obrázek 23: Domek, který je možno nakreslit jedním tahem.

Komentář:

Při kontrole úkolů dbejte spíše na správné použití bloků než na přesnost vzdáleností a úhlů. Hodnoty by však měly alespoň přibližně dávat smysl. Kromě funkčnosti kontrolujte i použité ikony a názvy jak u celých bloků, tak u jednotlivých parametrů. Diskutujte s žáky, proč právě tento název či tuto ikonu zvolili. Není třeba, aby použití vždy dávalo smysl vám, musí hlavně dávat smysl žákům. Rozlišujte však vychytralé žáky, kteří si umí zdůvodnit i očividný nesmysl.

Dostáváme se už ke komplikovanějšímu programování. Žák má nejprve obecně programovat jednotlivé funkce s parametry a až poté celý program. Doporučujte žákům zkoušet už samotné části programu a zkoušet je s různými parametry. Je velmi důležité mít odladěné jednotlivé části, než budou použity ve výsledném programu. Pro větší názornost můžete zkusit k robotu připevnit tužku nebo fix, aby byla ujetá dráha poté vidět na podložce, po které robot jel.

Domek z pátého úkolu lze nakreslit jedním tahem pouze v případě, že začínáte v jednom z dolních rohů. Nejprve nechte žáky, ať na papír nakreslí domek vlastnoručně jedním tahem. Nakreslit domek jedním tahem bude pro některé žáky komplikované. Diskutujte s žáky, za jakých podmínek je nakreslení domku

jedním tahem možné. Teprve až žáci toto zvládnou, mohou se pustit do programování. Délka šikmých (úhlopříčných) čar je iracionální číslo. Se staršími, pokročilejšími či nadanými žáky diskutujte, jak spočítat délku šikmých čar s využitím Pythagorovy věty. Je možné to udělat přímo v programu. Pro většinu žáků však bude stačit, když délky šikmých čar odhadnou.

2.2.17 Tvorba bloků s výstupními parametry

Cíl hodiny:

Žák se naučí vytvářet vlastní bloky s výstupními parametry, které pak dále může použít v programu.

Předchozí znalosti:

Žák zná příkazy k ovládání robotu, čtení senzorů a řízení běhu programu. Dále umí vytvářet vlastní bloky, u kterých zvládne nastavit vstupní parametry.

Potřebné pomůcky:

- Základní robot (Driving Base)
- Gyroskopický a ultrazvukový senzor

Úkoly:

1. Vytvořte program, ve kterém robot hledá nejbližší objekt.
   - Robot se nejprve otočí o 360 °.
   - Během otáčení robot hledá nejbližší objekt a zapamatuje si jeho vzdálenost a úhel otočení mezi výchozím směrem a tímto objektem.
   - Zjištěnou vzdálenost a úhel program vypíše na displej.
2. Z celého programu vytvořte nový blok se dvěma výstupními číselnými parametry (vzdáleností nejbližšího objektu a úhlem otočení, ve kterém je objekt vidět). Po skončení hledání je třeba zapamatované hodnoty vzdálenosti a úhlu otočení připojit na výstup. Hodnoty tentokrát není třeba vypisovat na displej.
3. Vytvořte program, ve kterém robot s pomocí nového bloku najde nejbližší objekt a přijede k němu (na vzdálenost 5 cm). Pro otočení a cestu k předmětu můžete použít bloky z minulé hodiny.
4. Podle bloku vytvořeného v bodě 2 vytvořte další vlastní blok. Bude mít navíc jeden vstupní a jeden výstupní parametr. Vstupním parametrem bude maximální vzdálenost, kterou má ověřovat, a výstupním parametrem bude logická hodnota, jestli byl v kruhu o daném poloměru nějaký objekt nalezen, či nikoli. Další dva výstupní parametry zůstanou stejné.

5. Upravte program z úkolu 3 s použitím bloku vytvořeného v úkolu 4. Robot opět hledá nejbližší předmět. Pokud předmět najde, přijede k němu. Pokud nic nenajde, na displeji se objeví nápis „volno“. Vyzkoušejte program s různými poloměry prohledávaného kruhu a různě vzdálenými objekty.

Komentář:

Kontrolujte správnost algoritmu pro nalezení minima. Pokud žáci na algoritmus nepřijdou sami, zkuste jim mírně poradit. Pokuste se poradit jen částečně, nikoli vymyslet algoritmus za ně. Ověřování vzdálenosti by mělo být spojité. Samotné natočení a cesta k předmětu je snadný úkol. Kontrolujte, zda žáci opravdu v programu používají svůj vytvořený blok. Stejně tak i při kontrole podmínky nalezení objektu.

Pokud naleznete v programu chybu, nemusíte o ní žákům hned říkat. Můžete místo toho zkusit namodelovat situaci, ve které se chyba projeví, a nechte žáky samostatně odhalit, co se stalo a proč. Při tvorbě bloku v úkolu 4 je možné zkopírovat tělo bloku z druhého úkolu a přidat potřebné podmínky nebo je možné uvnitř bloku použít dříve vytvořený blok. Obě možnosti jsou správně, je však dobré, aby žáci věděli zejména o druhé zmiňované možnosti. Pokud tedy zvolili možnost první, o té druhé jim alespoň řekněte. Jde tu o fakt, že vlastní bloky lze bez problému používat i uvnitř dalších vlastních bloků, nejen v hlavním programu.

2.2.18 Náhodná čísla

Cíl hodiny:

Žák pozná příkaz pro generování náhodných čísel.

Předchozí znalosti:

Žák zná příkazy pro ovládání robotu, nekonečnou smyčku a zpravocání dat z ultrazvukového senzoru.

Potřebné pomůcky:

- Základní robot (Driving Base) (2×)
- Ultrazvukový senzor

Úkoly:

1. V první části vytvořte cvičný cíl. Robot se bude pohybovat naprosto nepředvídatelně, tedy náhodně.

- Rychlost (případně směr) pohybu se bude měnit každou sekundu. Rychlost (resp. síla motorů) se může pohybovat v intervalu od -50 % do 50 %.
- Zajistěte, aby robot nevyjel z daného rozsahu. Od výchozího místa se může vzdálit maximálně o nějakou danou vzdálenost, aby nahodile nedojel až na okraj místnosti.

2. Dále vytvoříme druhý program. Robot bude s použitím ultrazvukového senzoru udržovat stálou vzdálenost od objektu před ním. Pokud se objekt přiblíží, robot couvne. Pokud se objekt oddálí, robot jede vpřed, jinak stojí na místě a čeká.

3. Vyzkoušejte funkčnost druhého programu s použitím dvou robotů. Jeden bude dělat cvičný (náhodně se pohybující) objekt a druhý bude udržovat danou vzdálenost.

Komentář:

V prvním úkolu ověřte, že se robot pohybuje opravdu náhodně. Někteří žáci možná nerozpoznají souvislost mezi náhodným číslem a náhodným pohybem, takovým můžete po chvíli snažení poradit konkrétní použití příkazu pro náhodná čísla.

Ve druhé části se bude druhý robot při pomalém pohybu cíle pohybovat trhaně. Vyžadujte po žácích větší plynulost pohybu. Pokud nikdo nepřijde na to, že mohou rozdělit pohyb na více intervalů podle vzdálenosti, můžete jim tuto myšlenku poradit. Neraďte jim však, jak to provést. To už by měli zvládnout sami. Mezi intervaly jednotlivých rychlostí mohou být také prázdné intervaly, které rychlost nijak neovlivní. V těchto prázdných intervalech se v podstatě sousední intervaly jednotlivých rychlostí překrývají. To opět o trochu zlepší plynulost pohybu.

Ve třetím úkolu budou muset žáci spolupracovat, aby měli k dispozici dva roboty. Je však dobré, aby byly na oba roboty vyzkoušeny programy jednoho i druhého žáka (resp. skupiny žáků). Zajímavé pro žáky může být i vzájemné porovnání jejich programů a případně zhodnocení nedostatků nebo předností jednoho či druhého.

2.2.19 Náhodná čísla podruhé

Cíl hodiny:

Žák si procvičí příkaz pro generování náhodných čísel při tvorbě složitějšího programu (hry).

Předchozí znalosti:

Žák umí generovat náhodná čísla, zvládne použít příkaz pro smyčku (cyklus) na

daný počet opakování. Zná příkazy pro ovládání robotu a umí použít příkaz pro čekání na různé senzorické události.

Potřebné pomůcky:

- Základní robot (Driving Base)
- Senzor barev
- Vyznačené barevné čáry na zemi (např. barevná lepicí páska, 3 barvy)

Úkoly:

1. Vytvořte hru podle následujících pravidel:
    - Na podložce jsou vyznačené barevné čáry. Robot se přes ně pohybuje po tazích.
    - Na začátku tahu robot čeká na kliknutí na prostřední tlačítko kostky.
    - Po kliknutí (stisk a uvolnění) robot půl sekundy počká, „hodí kostkou“ (vygeneruje náhodné číslo od jedné do šesti a vypíše ho na displej) a rozjede se vpřed.
    - Na každé čáře se robot na půl sekundy zastaví. Robot může popojet o tolik čar, kolik hodí na kostce.
    - Po skončení tahu robot čeká na stisk tlačítka a zahájení tahu dalšího.
    - Černé čáry znázorňují normální políčka. Zelená čára znamená cíl. Robot, který dojel na zelenou čáru vyhrál. Neměl by pokračovat dál v jízdě ani v případě, že ještě neprojel příslušný počet políček daného tahu.
    - Červená čára je past. Když na červené čáře robot skončí tah, musí se vrátit na začátek (musí si tedy pamatovat, jak daleko zatím dojel).

Komentář:

Čáry na zem lepte kolmo ke směru pohybu robotu. Mohou být libovolně daleko od sebe, ale myslete na to, aby jich bylo dostatek. Kolem dvaceti čar by mohlo stačit, ale může být i více. Čím více čar bude, tím zajímavější bude hra.

Poslední čáru nalepte páskou jiné barvy (zelená), aby mohl robot poznat, že je v cíli. Pokud nemáte jinou barvu, změňte zadání třeba na předem daný celkový počet přejetých čar. Další barvou (červená) udělejte několik málo čar, které budou znázorňovat pasti. V programu je rozdíl mezi ověřováním červené a zelené. Zatímco cíl je nutno ověřovat po každém políčku, pasti se naopak musí ověřit pouze na konci tahu. Kontrolujte, zda žáci provádí obě ověřování na příslušných místech programu.

Když uděláte čáry dostatečně dlouhé, bude dráha širší a může hrát více robotů naráz. Až budou žáci hotovi, můžete si hru zahrát. Žáci budou střídavě

podle tahů klikat na své roboty a fandit jim. Nutno však upozornit, že výsledek hry závisí zcela na náhodě, nikoli na kvalitě vytvořeného programu. Robot, který nějakým způsobem poruší pravidla hry může být vyloučen či penalizován. Případné postihy však stanovte ještě před zahájením hry a v průběhu je již neměňte.

2.2.20 Komunikace robotů

Cíl hodiny:

Žáci se učí spolupráci. Zároveň se naučí používat příkazy pro komunikaci mezi více roboty.

Předchozí znalosti:

Žáci využijí program hry z minulé hodiny a ovládání robotu pomocí tlačítek kostky.

Potřebné pomůcky:

- Základní robot (Driving Base)
- Programovatelná kostka (druhá)
- Senzor barev
- Vyznačené čáry na zemi (dle možností stačí barevná lepicí páska, 3 barvy)

Úkoly:

1. Propojte robot pomocí Bluetooth s druhou programovatelnou kostkou, která bude v tomto programu sloužit jako ovladač.
    - Pokud na kostce (ovladači) stiskneme tlačítko nahoru nebo dolů, robot se rozjede dopředu nebo dozadu. Při stisku tlačítek vpravo nebo vlevo se robot na místě otáčí.
    - Při uvolnění tlačítka se robot zastaví.
    - Je možné stisknout více tlačítek naráz. Například při současném stisknutí tlačítek dopředu a vpravo robot jede dopředu a zároveň zatáčí doprava.
    - Současné stisknutí tlačítek vpravo a vlevo nebo nahoru a dolů samozřejmě nemá na robot žádný efekt.
2. Roboty spolu budou hrát hru z minulé hodiny. Tentokrát však na začátku tahu nečekají na stisk tlačítka, ale čekají, než dokončí svůj tah předchozí robot. Na konci svého tahu zase každý robot předá tah dalšímu robotu v pořadí.

Komentář:

V této hodině pravděpodobně nebudeme mít dvě kostky pro každého žáka, je žádoucí žáky spojit do dvojic (nebo větších skupinek podle potřeby). Žáci se tak učí nejen nastavovat komunikaci robotů, ale učí se zároveň komunikovat mezi sebou. Některé skupiny možná budou potřebovat se vzájemnou komunikací pomoci. Zkuste jejich komunikaci usměrnit či zprostředkovat, nikoli však ji úplně vést.

V prvním úkolu pro ovládání robotu bude možná pro některé žáky obtížné naprogramovat sčítání efektu tlačítek. Tuto část jim můžete odpustit. V této hodině je důležité, aby si žáci vyzkoušeli komunikaci mezi více roboty, nikoli aby se zabývali složitějším řízením motorů.

Při komunikaci přes Bluetooth dávejte pozor na pojmenování robotů. Každý robot v místnosti by měl být pojmenovaný jinak. Kontrolujte pak, že pojmenování robotu v programu odpovídá s pojmenováním robotu, kterému chcete zprávu poslat. Robot může být v danou chvíli ve spojení pouze s jedním zařízením. K programování robotů proto bude možná jednodušší použít USB kabel. Umožníte tak zachovat spojení přes Bluetooth mezi roboty navzájem.

2.2.21 Pole

Cíl hodiny:

Žák se seznámí s pojmem pole hodnot a naučí se ho použít.

Předchozí znalosti:

Žák umí pracovat s čísly a proměnnými. Dokáže číst tlačítka kostky.

Potřebné pomůcky:

- Základní robot (Driving Base)

Úkoly:

1. Prozkoumejte proměnné typu Numeric Array (číselné pole). Jaké hodnoty do něj můžeme zapsat?

2. Vytvořte program, který vypíše na displej všechny prvky pole. Použijte cyklus (Loop) příkaz Array Operations – Read at Index. Vyzkoušejte výpis různých polí, například [1], [ ], [0,3,7,6].

3. Z předchozího programu udělejte nový blok s jedním vstupním parametrem, kterým bude právě pole čísel.

4. Vzpomeňte si na program, ve kterém jsme robotu zadávali příkazy pomocí tlačítek na kostce. Ten dnes vylepšíme, aby robot mohl provádět posloupnost více příkazů najednou. Zadané příkazy si robot postupně ukládá do paměti (do číselného pole). K dispozici bude nabídka čtyř příkazů:

    - Tlačítko nahoru – pohyb vpřed.
    - Tlačítko dolů – pohyb vzad.
    - Tlačítko doprava – otočení doprava.
    - Tlačítko doleva – otočení doleva.
    - Prostřední tlačítko – provedení příkazů. Robot počká půl sekundy a poté postupně provede všechny zadané příkazy. Po vykonání posledního příkazu program pokračuje od začátku.

Komentář:

Použití polí je náročná záležitost. Žáci musí pochopit zcela nový princip a používat několik nových příkazů. Kontrolujte jejich práci průběžně, aby se zbytečně nezdrželi na drobných chybách. Pro práci s polem musí žáci zvládnout čtení a zápis proměnných a speciální operace s polem, ale také se musí dobře orientovat v použití podmínek a cyklů. Při řešení úloh mnoho žáků zapomene počítat s prázdným polem. Další častou chybou je nepochopení rozdílu mezi prázdným polem a polem s nulou. Toto je nutné jim případně vysvětlit.

Důsledně kontrolujte obzvláště program z úkolu číslo 2. Výpis pole testujte na různých hodnotách. Vyzkoušejte dostatečně velké pole, prázdné pole, pole se záporným i desetinným číslem, pole s nulou a další. Úkol číslo 4 je zaměřen na příkaz Append.

V této hodině bude možná užitečné ladit program za běhu pomocí aplikace v počítači. Jako alternativu bodu přerušení (breakpointu) můžeme využít přidané příkazy Wait. Za běhu programu můžeme sledovat hodnoty jednotlivých propojek mezi příkazy. Program bohužel není možné krokovat po částech.

2.2.22 Logika

Cíl hodiny:

Žák se seznámí s typem logických hodnot a naučí se ho použít.

Předchozí znalosti:

Žák zvládá řídit robot, číst hodnoty senzorů, pracovat s proměnnými a používat větvení.

Potřebné pomůcky:

- Základní robot (Driving Base)

- Senzor barev, kabel a konstrukční díly k připojení
- Různobarevné lepicí pásky, kousky barevného papíru (červená, žlutá, zelená, modrá)

Úkoly:

1. Podle návodu v aplikaci připojte senzor barev směrem dolů.
2. Vytvořte program, ve kterém robot popojede o jednu otočku motoru vpřed, pokud barevný senzor při spuštění programu viděl červenou barvu, jinak popojede stejně, ale směrem dozadu.
3. Toto se dá udělat nejméně dvěma způsoby. Příkazu Switch lze nastavit podmínka přímo podle senzoru barev. Je také možné použít žlutý blok pro porovnání čtené barvy (Colour Sensor – Compare – Colour) a výstup napojit na Switch s obecnou logickou podmínkou. Vyzkoušejte oba způsoby.
    - Už jsme se setkali s hodnotami číselnými a textovými. Existuje ještě další typ hodnoty – logická. Zatímco text může být libovolná posloupnost znaků a číslo může mít libovolnou velikost, logická hodnota má pouze dvě možnosti: pravda (true), nebo nepravda (false). Tyto hodnoty mohou být například výsledkem různých porovnání.
4. V dalším programu robot popojede vpřed o dvě otočky motoru. Poté se otočí a vrátí zpět. Pokud však někde během cesty viděl červenou barvu, vracet se nesmí, pouze se otočí čelem vzad a skončí. Musí si tedy pamatovat, zda červenou viděl nebo nikoli.
5. Předchozí program upravte s použitím logických spojek:
    - Robot se smí vrátit, pokud viděl červenou nebo zelenou.
    - Logické spojky se dají řetězit za sebe. Pokud cestou robot viděl červenou i zelenou, nebo alespoň modrou, před případným návratem se otočí kolem dokola.
    - Pokud viděl červenou, nebo zelenou, nikoli však obě naráz, bliká při návratu oranžové světlo.
6. Jednotlivé barvy můžete ukládat v několika logických proměnných, nebo v jednom logickém poli na indexech podle čísla barvy. Naprogramujte úlohu 6 také druhým způsobem. Některou část kódu to pravděpodobně zjednoduší, jinou naopak zkomplikuje. Které části to jsou?
7. Všechny možné hodnoty a jejich kombinace pro různé logické spojky ukazuje tabulka 1. Hodnoty pravda (true) a nepravda (false) se někdy značí jako 1 (true) a 0 (false). Tabulku prozkoumejte a doplňte hodnoty následujících výrazů a navzájem je porovnejte.

- c = (a AND b) OR ((NOT a) AND (NOT b))
- d = NOT (a XOR b)

a	b	NOT a	a AND b	a OR b	a XOR b	c	d
0	0	1	0	0	0		
0	1	1	0	1	1		
1	0	0	0	1	1		
1	1	0	1	1	0		

Tabulka 1: Všechny kombinace logických hodnot pro různé logické spojky

Komentář:

Některé úkoly lze splnit bez použití logických hodnot, proměnných a spojek. Dohlédněte na žáky, aby logické hodnoty uměli použít. Chvalte je i za jiná řešení, ale cílem této hodiny je naučit se používat logické hodnoty a spojky. Úkoly v této hodině jsou spíše jednodušší. Úkolů je více, žáci určitě nemusejí stihnout všechny, ale měli by zvládnout použití logických spojek.

Programy kontrolujte na co nejvíce kombinacích barev, které robot může projet. Kontrolujte nejen funkčnost, která může být závislá na konkrétním zvoleném testu, ale hlavně správnost programu revizí zdrojového kódu. Pokud odhalíte chybu, nemusíte ji žákovi hned ukazovat. Spíše namodelujte situaci, na které se chyba projeví a nechte pak žáka na chybu přijít samostatně. Pokud se mu to nepovede, až po nějaké chvíli mu chybu ukažte, ale neopravujte ji za něj.

U spojky nebo budou mít nejspíše někteří žáci problém s rozlišením disjunkce klasické (OR) a exkluzivní (XOR). Při kontrole úkolů se na správné použití těchto dvou spojek obzvláště zaměřte, jelikož není vidět na první pohled, ale jedná se o častou chybu.

2.2.23 Uklízející robot

Cíl hodiny:

Žák naprogramuje robot, který uklízí předměty kolem sebe. Průprava pro následující hodinu.

Předchozí znalosti:

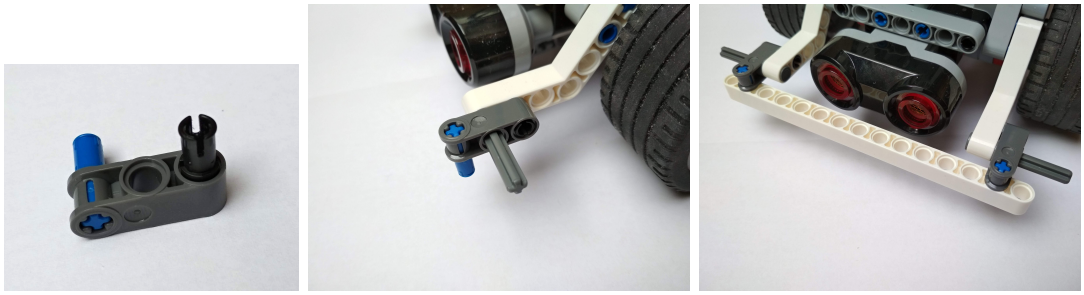
Žák zvládá příkazy pro řízení robotu a umí číst data z ultrazvukového senzoru.

Potřebné pomůcky:

- Základní robot (Driving Base)
- Ultrazvukový a gyroskopický senzor
- Konstrukční díly a kabel k připojení

Úkoly:

1. Podle návodu v aplikaci připojte k robotu ultrazvukový senzor.
2. Přidejte k robotu nárazník (viz obrázek 24) tak, aby chránil ultrazvukový senzor, ale nepřekážel mu ve „výhledu“.



Obrázek 24: Návod k sestrojení nárazníku před ultrazvukový senzor.

3. Naprogramujte robot, který vyčistí kolem sebe kruhový prostor o poloměru přibližně 50 cm.
   - Robot se otáčí a jakmile spatří předmět, který se nachází blíž než 50 cm, rozjede se proti němu a odtlačí ho do vyhovující vzdálenosti.
   - Poté se robot vrátí na výchozí pozici a pokračuje v kontrole.
   - Program končí, když se robot otočí kolem dokola a vyčistí celý kruh.

Komentář:

Tato hodina by měla být pro žáky snazší. Ke splnění úkolů není zapotřebí žádné složitější myšlenky ani použití složitějších příkazů. Samotný program bude spíše menšího až středního rozsahu. Tato hodina však slouží jako příprava přespříští (složitější) hodiny.

Robot se nepohybuje přesně. Může se tedy stát, že se při návratu po odtlačení překážky nevrátí přesně na stejné místo, ze kterého začínal. Pokud bude čas, pokuste se tento návrat vyladit použitím vhodných parametrů u příkazů. Nepřesnost v této hodině sice ničemu nevadí, ale v přespříští hodině, kde budeme tento program využívat, už by větší nepřesnosti vadit mohly.

Pokud to žáky nenapadne, připomeňte jim, že je možné použít blok pro ujetí dané vzdálenosti, který tvořili v dřívějších hodinách. Pomůže jim to zpřehlednit program a urychlit jeho tvorbu.

2.2.24 Řazení čísel

Cíl hodiny:

Žák se naučí seřadit pole čísel.

Předchozí znalosti:

Žák umí pracovat s polem, generovat náhodná čísla a vypsat výstup na displej, používat cykly a větvení, najít nejmenší prvek, porovnat hodnoty. Zvládá příkazy pro řízení robotu.

Potřebné pomůcky:

- Základní robot (Driving Base)
- Vyznačená čísla na podložce

Úkoly:

1. Vytvoříme robot, který umí seřadit čísla podle jejich velikosti:
    - Program vytvoří číselné pole a do něj vloží deset náhodných čísel od 0 do 9 a vypíše je na displej.
    - Na podložce jsou vyznačena čísla od 0 do 9. Robot začíná na nule. Podle prvního čísla v poli dojede na příslušné číslo na podložce, tam zabliká oranžovým signálním světlem a vrátí se zpět. Stejně pokračuje pro všechna čísla v poli.
    - Po vyčerpání všech čísel se robot zastaví a čeká pět sekund.
    - Čísla v poli se vzestupně seřadí a vypíšou se na displej pod původní čísla.
    - Robot opět jezdí po číslech na podložce podle čísel v seřazeném poli. Na každém čísle se zastaví, oranžově zabliká a vrátí se zpět.

Komentář:

Zadání této hodiny je velmi jednoduché. Provedení však už tak snadné nebude. Pro řazení čísel máme mnoho algoritmů. I pro jednodušší řadicí algoritmy budou programy tvořené v blocích patřit spíše mezi ty složitější.

Nejintuitivnější (tedy nejjednodušší na pochopení) bude nejspíše řazení výběrem (Select sort): Program vždy najde nejmenší prvek pole, ten přidá na konec nového seřazeného pole a ze starého pole jej smaže. Další podobně jednoduchý algoritmus je řazení vkládáním (Insert sort), který postupně vybírá z původního pole hodnoty a zařazuje je do nového seřazeného pole. Třetí možností je algoritmus řazení výměnou sousedních dvojic (Bubble sort): prochází opakovaně pole a prohazuje jednotlivé sousední hodnoty, které jsou navzájem ve špatném pořadí.

Můžete s žáky diskutovat více algoritmů a vybrat ten nejvhodnější. Insert sort nejspíš nebude možné použít, jelikož v Mindstorms existuje pouze příkaz pro vložení nového prvku až na konec pole. Select sort a Bubble sort jsou pro nás rovnocenné. Každý má několik výhod i nevýhod.

Značky pro jednotlivá čísla na zemi umístěte pravidelně. Při najíždění na tyto značky nemusíte kontrolovat přesnost. Pokud robot vyjede bokem nebo nějakou značku přejede, ničemu to nevadí. V této hodině je důležitější pochopení a naprogramování řazení čísel.

2.2.25 Řazení podruhé

Cíl hodiny:

Žák si procvičí a prohloubí dovednosti použití algoritmů pro řazení pole hodnot.

Předchozí znalosti:

Žák dokáže pole seřadit, zvládá příkazy pro ovládání robotu a čtení hodnot měřených ultrazvukovým senzorem.

Potřebné pomůcky:

- Základní robot (Driving Base)
- Ultrazvukový a gyroskopický senzor
- Konstrukční díly a kabely k připojení

Úkoly:

1. Vytvoříme alternativu uklízejícího robotu:
    - Robot se nejprve otočí kolem dokola a zapamatuje si všechny překážky. Je nutné si pro každou překážku zapamatovat její vzdálenost a úhel otočení.
    - Jakmile robot „obhlédne“ své okolí, seřadí překážky podle vzdálenosti.
    - Pro přehlednost programu používejte vlastní bloky. Zamyslete se, jaké budou mít vstupní a výstupní parametry.
    - Robot bude odstraňovat překážky postupně od té nejbližší až po tu nejvzdálenější.
2. Zajistěte, aby se robot otáčel vždy takovým směrem, aby otočení mělo co nejmenší velikost úhlu.

Komentář:

Žák již sice umí seřadit pole hodnot, ale v této hodině má řadit dvě pole podle hodnot v jednom z nich. Veškerá porovnávání tedy provádí pouze v jednom poli, ale zápis prvků musí dělat s oběma poli. Hodina bude nejspíš velmi náročná, a to i časově. Pravděpodobně ji bude potřeba rozšířit do více hodin.

Vzhledem k náročnosti zadání netrvejte na vyřešení všech úkolů, je možné zadání žákům v případě potřeby upravit či zjednodušit. Jedním z možných zjednodušení je místo řazení polí pouze najít v polích nejbližší předmět a ten odstranit. Až pokud se toto žákům povede, mohou daný předmět z pole odstranit (resp. nastavit vzdálenost na větší číslo) a opět hledat nejbližší předmět. Nakonec odklidí postupně všechny předměty v pořadí podle jejich vzdálenosti, čímž vlastně splní původní zadání.

Pokud žáci samostatně zvolí postupné odklízení zmíněné v předchozím odstavci a budou rychle hotovi, pochvalte je za správné a rychlé splnění úkolu. Dále je možné po nich vyžadovat, aby jednotlivá pole byla nejprve celá seřazená a až poté robot začal s odklízením.

Poslední úkol s kontrolou kratšího úhlu nemusí žáci nutně stihnout. K jeho splnění je třeba trocha počítání a přemýšlení. Pro nadanější žáky, kteří budou dřív hotovi, však může být zajímavou výzvou.

2.2.26 Stavba třídicího robotu

Cíl hodiny:

Žák postaví třídicí robot podle návodu.

Předchozí znalosti:

Žák zná součástky Mindstorms, umí je používat, dokáže pracovat podle návodu.

Potřebné pomůcky:

- Konstrukční a robotické díly podle návodu

Úkoly:

1. Rozeberte modely postavené v předchozích hodinách a roztřiďte díly na příslušná místa.

2. Spusťte aplikaci a otevřete návod k sestavení třídiče barev (Building Instructions – Core Set Models – Colour Sorter).

3. Postupujte podle návodu a třídič sestavte.

Komentář:

Úkoly v této hodině budou velmi snadné. Otevřít návod a postupovat podle něj by měl jistě zvládnout každý žák. Tato hodina je však časově náročná. Stavba robotu zabere celou hodinu (90 min). Je však možné, že někteří žáci robot dokončí o něco dříve, jiným však nemusí celá hodina stačit. Uvidíte, jak rychle budou žáci postupovat. Rychlejší mohou třeba po skončení pomoci pomalejším. Nebo

mohou dostat individuální zadání, případně pokračovat zadáním další hodiny, které zahrnuje práci se sestaveným robotem a jeho zdrojovým kódem.

2.2.27 Třídicí robot

Cíl hodiny:

Žák zvládá porozumět cizímu programu.

Předchozí znalosti:

Žák v minulé hodině sestavil robot pro třídění barev. Zná většinu základních příkazů pro ovládání robotu, čtení senzorů a řízení programu (proměnné, podmínky, cykly).

Potřebné pomůcky:

- Robot pro třídění barev
- Barevné nosníky délky 3 cm různých barev
- Plastové kelímky nebo jiné nádobky

Úkoly:

1. Návod k sestavení třídicího robotu obsahuje i program k jeho ovládání. Vyzkoušejte, jak robot funguje.
2. Projděte si pečlivě kód programu a zjistěte, k čemu která část kódu slouží. Své poznatky si do kódu poznamenejte v podobě komentářů.
3. Když se zadá robotu určitý počet kostek, robot sám začne kostky třídit i bez stisknutí tlačítka. Jaký je tento počet? Najděte v programu část kódu, která tento maximální počet nastavuje a upravte jej na maximálně 5 kostek.
4. Program nyní třídí barvy na 4 různá místa. Upravte jej tak, aby třídil pouze na dvě různá místa (modrá a zelená společně a žlutá s červenou společně).
5. Umožněte robotu přijímat také bílou. Bílé dílky budou roztříděny na třetí místo.
6. Vytvořte dva vlastní bloky:
   - První bude zadávání barevných kostek. Bude mít jeden vstupní parametr pro maximální počet kostek a jeden výstupní parametr pro pole zadaných barev.
   - Další vlastní příkaz bude třídění kostek s jedním vstupním parametrem, kterým bude pole zadaných barev k roztřídění.

7. Nově vytvořené bloky použijte v programu místo příslušných částí kódu. Program pak bude o něco přehlednější.

Komentář:

S prvním úkolem žákům může pomoci video na začátku návodu. Nejspíš na to přijdou sami. Je však potřeba, aby na prvním úkolu neztratili příliš času. Druhý úkol je klíčový, na úspěchu u tohoto úkolu závisí úspěch úkolů ostatních. Další úkoly spíše ověřují pochopení kódu tím, že žák musí najít část kódu implementující danou část chování robotu. Úprava nalezené části není náročná.

Opět je těžké předvídat, kolik času budou žáci k řešení úkolů (obzvlášť úkolu číslo 2) potřebovat. Navíc někteří budou ještě pravděpodobně dokončovat stavbu robotu. Tuto (i předchozí) hodinu je možné libovolně roztáhnout do více hodin podle situace. Úkoly jsou řazeny od základních k pokročilým. První čtyři úkoly by měli zvládnout všichni žáci. Další jsou spíše rozšiřující pro rychlejší a nadanější žáky.

2.2.28 Stavba balancujícího robotu

Cíl hodiny:

Žák postaví balancující robot podle návodu.

Předchozí znalosti:

Žák zná součástky Mindstorms, umí je používat, dokáže pracovat podle návodu.

Potřebné pomůcky:

- Konstrukční a robotické díly podle návodu

Úkoly:

1. Rozeberte modely postavené v předchozích hodinách a roztřiďte díly na příslušná místa.
2. Spusťte aplikaci a otevřete návod k sestavení balancujícího robotu (Building Instructions – Core Set Models – Gyro Boy).
3. Postupujte podle návodu a robot sestavte.

Komentář:

Úkoly v této hodině jsou snadné. Naopak náročnější bude časová koordinace. K sestavení poměrně složitého balancujícího robotu pravděpodobně nebude žákům jedna hodina (90 min.) stačit. Někteří žáci budou opět rychlejší, jiní pomalejší. Rychlejší mohou pomoci spolužákům nebo pokračovat v úkolech dalších

hodin. Nemá smysl na ostatní spolužáky čekat. Až rychlejší žáci vyřeší i všechny úkoly následující hodiny, mohou se věnovat vlastnímu projektu nebo začít stavět zápasnický robot (viz hodinu 2.2.30).

2.2.29 Balancující robot

Cíl hodiny:

Žák se naučí číst složitý kód. Pozná kód jednotlivých částí a určí jejich funkci bez nutnosti porozumět detailům implementace daných částí.

Předchozí znalosti:

Žák zná součástky Mindstorms, umí je používat, dokáže pracovat podle návodu.

Potřebné pomůcky:

- Sestavený balancující robot
- Programovatelná kostka (druhá)

Úkoly:

1. Naučte se používat balancující robot podle kódu přiloženého k návodu.
2. Projděte si kód. Jistě si všimnete vysoké míry jeho složitosti. V kódu jsou navíc použity vlastní bloky. Podívejte se zběžně dovnitř, co tyto bloky dělají.
3. Není nutné kódu rozumět do detailu. Spíše si kód pomyslně rozdělte na jednotlivé části a určete, k čemu která část slouží.
4. Poznatky si do kódu poznamenejte ve formě komentářů.
5. Zejména určete části, které slouží k udržení rovnováhy a k ovládání robotu (zadání příkazu).
6. Upravte program tak, aby bylo možné robot ovládat přes Bluetooth pomocí tlačítek jiné kostky:
   - Kliknutí na tlačítka nahoru/dolů spustí pohyb robotu vpřed/vzad.
   - Kliknutí na tlačítka doprava/doleva spustí otáčení robotu.
   - Kliknutí na prostřední tlačítko robot zastaví.
7. Upravte předchozí program následovně:
   - Stisknutí tlačítek spustí pohyb/otáčení robotu.
   - Uvolnění tlačítek pohyb zastaví.

- Prostřední tlačítko nemá žádný efekt.
- Efekt tlačítek pohybu a otočení se může sčítat.

Komentář:

Použití balancujícího robotu žáci objeví sami. Pomůže jim v tom video na prvním snímku návodu. Vyznat se v kódu přiloženého programu bude naopak velmi náročný úkol. První část kódu sloužící k inicializaci robotu a udržení rovnováhy je velmi složitá. Druhou část kódu by žáci alespoň přehledově pochopit měli. Toto pochopení je důležité pro splnění dalších úkolů. Klíčový fakt k pochopení druhé části je, že robot se zde vůbec neovládá bloky pro řízení motorů, nýbrž pouhým nastavením proměnných, které pak mají účinek na vykonání příkazů řízení motorů v první části kódu. Trvejte na vytvoření komentářů v kódu. Ústní vysvětlení nestačí. Aby mohl žák formulovat smysluplnou písemnou větu, musí lépe proniknout do pochopení kódu.

Úkol pro ovládání robotu pomocí druhé kostky bude v případě pochopení příslušných částí kódu snadný. Jedná se pouze o použití bloků pro čtení zpráv, které už žáci znají z dřívějška. K vyzkoušení si mohou půjčit kostku od spolužáků. Mohou si půjčit celý robot, jehož je kostka součástí, a nemusí jej ani rozebírat. Stačí do kostky nahrát svůj program. Cílem úkolu je ověřit, že žák pochopil, která část programu zajišťuje vnitřní mechanismy a která skutečně ovlivňuje činnost robotu.

Poslední úkol je pouze komplikovanější variantou úkolu předchozího. Je určen pro nadané a rychlejší žáky. Nemusí ho stihnout všichni.

2.2.30 Robotí zápasy

Cíl hodiny:

Žák rozvíjí samostatnost tvorbou komplexního programu podle volného zadání.

Předchozí znalosti:

Žák umí používat příkazy pro řízení robotu, zvládá číst data ze senzorů a zná většinu hlavních programových konstrukcí.

Potřebné pomůcky:

- Programovatelná kostka
- Všechny dostupné motory a senzory
- Sada konstrukčních dílů ke stavbě robotu
- Vyznačený zápasnický ring (kruh na podložce, např. barevnou páskou)

Úkoly:

1. Postavte zápasnický robot:
    - Ke stavbě robotu můžete použít libovolné součástky, motory a senzory z jedné sady.
    - Robot můžete založit třeba na základním robotu (Driving Base), ale není to nutné.
    - Pamatujte na ochranu senzorů, aby během zápasu nedošlo k jejich poškození.
2. Naprogramujte robot tak, aby zápasil podle následujících pravidel:
    - Zápasí spolu vždy dva roboty.
    - Cílem je vytlačit soupeře z vyznačeného kruhu. Prohrává robot, který se jako první ocitne celý mimo kruh.
    - Na začátku se umístí oba roboty do kruhu zády k sobě a čekají na stisk prostředního tlačítka kostky.
    - Po stisknutí tlačítka robot počká 5 sekund a poté začne zápasit.
    - Mezi jednotlivými zápasy turnaje se roboty nesmějí mechanicky měnit, mohou se pouze opravit do původního stavu v případě poškození během zápasu.
    - Program robotů se mezi jednotlivými zápasy měnit může. Nesmí se však pohybovat s bloky, je možné pouze upravovat parametry jednotlivých bloků.

Komentář:

Žáci budou potřebovat alespoň jednu hodinu k sestavení robotu. Další hodinu můžou dostat k programování a zkoušení zápasů mezi sebou. Ve třetí hodině můžete uspořádat turnaj. Počet hodin je možné libovolně upravit podle potřeby.

Ring nemusí být přesně kruhový, s rovnou lepicí páskou to ani není možné. Použijte libovolný mnohoúhelník. Je však doporučeno, aby se jednalo minimálně o šestiúhelník a aby byl pokud možno pravidelný.

Pro motivaci je možné žákům ukázat nějaké video zápasů.[3]

Při tvorbě a programování robotů je možné žákům poradit či odpovědět na otázky. Dbejte však na to, abyste některé žáky nezvýhodňovali příliš dobrou radou. Připomínejte žákům, aby si i v průběhu práce svá řešení ukládali pro případ nehody. Dbejte o to, aby žáci udržovali svůj zdrojový kód přehledný. Bude se pravděpodobně jednat o složité programy, důraz na přehlednost kódu je tedy klíčový.

[3] https://youtu.be/suyPkO0apak, https://youtu.be/-RY4DI9PUto

Závěr

Hlavním cílem této práce bylo vytvořit výukové materiály pro výuku robotiky s využitím stavebnice Mindstorms značky Lego. Výukové materiály byly vytvořeny v podobě učitelských pracovních listů obsahujících podrobné instrukce k přípravě a realizaci jednotlivých vyučovacích hodin. Pracovní listy obsahují cíle výuky, potřebné předchozí znalosti žáků, použité pomůcky, úkoly, které budou žáci v hodinách řešit, a v závěru komentář k praktickým aspektům přípravy a provedení výuky v každé hodině. Učitelské pracovní listy jsou dále doplněny vzorovými zdrojovými kódy ke všem úlohám a pracovními listy pro žáky obsahujícími pouze úkoly dané hodiny.

Výuka je koncipována pro žáky druhého stupně základních škol či odpovídajících ročníků víceletých gymnázií. Není však vyloučena účast ani mladších talentovaných žáků ani starších zájemců. Vytvořené pracovní listy nabízejí přes 30 vyučovacích hodin délky 60–90 minut, a tak pokryjí výuku po celý školní rok. Výukové materiály mohou být použity přímo na školách či v různých organizacích zabývajících se volnočasovými aktivitami pro děti a mládež. Výuka je koncipována spíše jako zájmový kroužek vzhledem k nedostatku prostoru pro robotiku či programování a algoritmizaci v klasických školních hodinách. S menšími úpravami by však bylo možné materiály použít i v klasické výuce.

Výukové materiály vytvořené v této práci jsou použitelné jak pro zkušené učitele robotiky k usnadnění přípravy hodin, tak i pro učitele začátečníky či nadšence, kteří se s robotikou nebo její výukou dříve nesetkali. Začínajícím učitelům přijdou vhod zejména úkoly jednotlivých hodin a jejich vzorová řešení. Ty je možné využít též jako studijní pomůcku, pro samostudium.

Obsah výuky je zaměřen na základní principy programování, algoritmizace a algoritmického myšlení, které jsou obecné a nezávislé na konkrétní platformě či programovacím jazyku. V hodinách robotiky se tak žáci naučí používat programové konstrukce jako porovnávání hodnot, podmínky, cykly, vytváření vlastních podprogramů nebo ukládání dat v proměnných či polích. Dále se žáci setkají se základními jednoduchými algoritmy pro vyhledávání a řazení.

Těším se na využití vytvořených výukových materiálů v praxi. V případě, že uvedené materiály využijete ve vlastní výuce, budu velmi rád za vaše zkušenosti i každou zpětnou vazbu. Tu mi, prosím, zašlete na mou osobní e-mailovou adresu tonda.vlach@seznam.cz.

Conclusions

The main goal of this thesis was to create a methodological material on teaching robotics using Mindstorms from Lego. The material was created as worksheets containing instructions to prepare and implement particular lessons. Worksheets consist of teaching goals, required previous pupils' knowledge, equipment needed, a few tasks for each lesson and finaly a comment describing some practical aspects of teaching the subject. Besides teacher's worksheets, there are sample source codes for every task and also worksheets for pupils. The pupils' worksheets contain only tasks for each lesson.

Teaching is designed for 11–15 years old pupils, but younger talented pupils or some older ones can attend these lessons too. The worksheets provide more than 30 lessons, therefore they cover whole school year. The lessons are 60–90 minutes long. The teaching material can be used in schools or any organization providing free time activities for children. Although the form of teaching is designed more as free time lessons in this thesis, with small adjustmensts, it can be used for classic school lessons too.

The teaching material created in this thesis can be used by both experienced robotics teachers, to make easier for them to prepare their lessons, and also by teachers beginning with robotics as a step by step guide. Tasks together with sample source codes can be used as a study material for such beginner robotics teachers.

The content of the lessons is designed to ephasize basic principles of programming, algorithmization and algorithmic thinking which are independent of any specific platform or programming language. So in prepared lessons, pupils learn how to use programming structures such as value comparisons, conditions, cycles, custom routines and saving data into variables or arrays. Pupils also meet some basic algorithms for searching and sorting.

I look forward to using the created material in actual teaching. In case you use the material, I would be grateful for any experience and feedback. You can send it to my personal e-mail address tonda.vlach@seznam.cz.

A Učitelské pracovní listy

Pracovní listy pro učitele stejně, jak jsou uvedeny v kapitole 2.2, jsou pro potřeby tisku přiloženy ještě v samostatném souboru PDF, který je k dispozici na přiloženém CD (viz přílohu D). Učitelské listy jsou zde uvedeny na samostatné straně (resp. dvojstraně) pro jednodušší orientaci. Soubor s učitelskými listy je dále doplněn krátkým přehledem jednotlivých vyučovacích hodin, který obsahuje kromě názvů hodin také potřebné znalosti a nové poznatky získané v každé hodině. Tento přehled by mohl pomoci lepší orientaci v materiálech jako celku.

B Žákovské pracovní listy

Žákovské pracovní listy jsou určené k tisku či elektronické distribuci pro žáky. Obsahují úkoly jednotlivých hodin. Pro žáky bude nepochybně dobré, když budou mít úkoly před sebou na papíře. Nebudou muset čekat, než jim učitel zadá další úkol, ale budou moci pokračovat svým tempem. Další výhodou tištěných úkolů je možnost zapisovat si do zadání vlastní poznámky. V takovém případě je ideální, když má žák možnost si pracovní list po skončení hodiny nechat pro případy zpětného nahlédnutí do vlastních poznámek v budoucnu.

Seznam úkolů v žákovských listech je stejný jako v listech učitelských. Učitel tedy žákovský list nepotřebuje.

Vzhledem ke shodnosti úkolů v žákovských a učitelských listech nemá smysl všechny žákovské listy uvádět do tohoto textu. Pro potřeby tisku nebo jiné distribuce jsou žákovské listy dostupné v souboru formátu PDF na přiloženém CD (viz přílohu D).

C Vzorové zdrojové kódy

Podstatnou částí práce jsou kromě pracovních listů také vzorové zdrojové kódy k úlohám uvedeným v pracovních listech. Jedná se tedy o jakési referenční řešení většiny úloh.

Vzorové zdrojové kódy jsou uvedeny téměř ke všem hodinám s výjimkou hodin 2.2.14, 2.2.26, 2.2.28 a 2.2.30. V hodině 2.2.14 žák ovládá robot mobilem, žádný kód tedy nevytváří. V hodinách 2.2.26 a 2.2.28 žák pouze staví nový robot dle návodu, opět tedy nevytváří žádný kód. V hodině 2.2.30 tvoří žák program samostatně dle volného zadání a neexistují ideální ani špatná řešení.

Všechny uvedené zdrojové kódy byly vytvořeny pro účely této práce s výjimkou kódů pro obsluhu třídicího a balancujícího robotu (v kapitolách 2.2.26–2.2.29), které byly převzaty z návodů dostupných v aplikaci Mindstorms EV3 Education, popř. mírně pozměněny či doplněny dle uvedených úkolů.

Uvedené zdrojové kódy však rozhodně nejsou jediným možným řešením. Je možné, že žáci najdou jiná správná řešení. V některých případech může být jiné nalezené žákovo řešení dokonce efektivnější nežli odpovídající vzorové řešení.

Vzorová řešení tedy nemají sloužit ke kontrole žákovských řešení, zda tomu vzorovému přesně odpovídají. Vzorové zdrojové kódy by spíše měly být pomůckou pro učitele při přípravě na hodinu. Při takové přípravě je dobré, aby si učitel každý zdrojový kód prošel a pochopil všechny jeho části. Předejde tak nepříjemnostem při hodině, které by mohly vzniknout, kdyby ani učitel nevěděl, jak některou úlohu řešit.

Vzorové zdrojové kódy jsou k dispozici na přiloženém CD (viz přílohu D). Nacházejí se zde v souborech formátu *.ev3, které je možné otevřít v aplikaci sloužící k programování Mindstorms (viz kapitolu 1.1.2).

D Obsah přiloženého CD

bin/
: Tato složka je prázdná. Předmětem práce nebyla tvorba žádného programu.

doc/
: Tento text diplomové práce v souboru typu PDF, dále zdrojové kódy pro sestavení tohoto textu práce v systému LaTeX a nakonec samostatné učitelské a žákovské pracovní listy.

src/
: Vzorové zdrojové kódy jako referenční řešení k jednotlivým úlohám. Zdrojové kódy jsou uloženy v souborech typu EV3, které je možné otevřít ve výše zmíněné aplikaci.

readme.txt
: Soubor s obsahem CD a instrukcemi pro jeho použití.

Literatura

[1] VALK, Laurens. *EV3 and NXT: Difference and Compatibility* [online]. 2013 [cit. 2020-2-11]. Dostupný z: ⟨http://robotsquare.com/2013/07/16/ev3-nxt-compatibility/⟩.

[2] HAVELKA, Martin; STOFFOVÁ, Veronika. *Robotika – stavba a programování robotů (LEGO Mindstorms EV3 a NXT)*. Olomouc: Univerzita Palackého, 2017. ISBN 978-80-244-5194-7.

[3] SOLDAAT, Xander. *Comparing the NXT and EV3 bricks* [online]. 2013 [cit. 2020-2-12]. Dostupný z: ⟨http://botbench.com/blog/2013/01/08/comparing-the-nxt-and-ev3-bricks/⟩.

[4] *Python for EV3*. [online]. [cit. 2020-2-13]. Dostupný z: ⟨https://education.lego.com/en-us/support/mindstorms-ev3/python-for-ev3⟩.

[5] LEVITIN, Anany; LEVITIN, Maria. *Algorithmic puzzles*. Oxford: Oxford University Press, 2011. ISBN 978-0-19-974044-4.

[6] STOFFOVÁ, Veronika; HAVELKA, Martin. *Práca s robotickými stavebnicami na 2. stupni ZŠ: zbierka riešených úloh*. Olomouc: Univerzita Palackého v Olomouci, 2018. ISBN 978-80-244-5383-5.

[7] PELÁNEK, Radek. *Programátorská cvičebnice: algoritmy v příkladech*. Brno: Computer Press, 2012. ISBN 978-80-251-3751-2.

[8] LEPIL, Oldřich. *Teorie a praxe tvorby výukových materiálů: zvyšování kvality vzdělávání učitelů přírodovědných předmětů*. Olomouc: Univerzita Palackého v Olomouci, 2010. ISBN 978-80-244-2489-7.

[9] SPRAUL, Anton. *Think like a programmer: an introduction to creative problem solving*. San Francisco, CA: No Starch Press, 2012. ISBN 978-1-59327-424-5.