1/2024, Volume 16, Issue 1
ISSN 1803-537X

SCRATCH SOFTWARE EXTENSION FOR THE LEGO MINDSTORMS ROBOTIC LEARNING TOOL

Patrik KLOFÁČ*, Jihočeská univerzita v Českých Budějovicích, Česká republika
Matěj PELIKÁN, Jihočeská univerzita v Českých Budějovicích, Česká republika

Přijato: 24. 1. 2024 / Akceptováno: 31. 5. 2024
Typ článku: Teoretická studie
DOI: 10.5507/jtie.2024.005

Abstract: The paper deals with the testing of the extension of the online block programming environment Scratch for Lego Mindstorms EV3 on tasks from the textbook Robotics with Lego Mindstorms for the 2nd grade of primary school created within the PRIM project (Support for the development of computational thinking). The textbook was written for software that is no longer used today. For these reasons, we decided to check whether all these tasks can be programmed in the block programming language Scratch and to present a few selected ones to the reader. We point out the facts how compatible the Scratch extension is with the building block, which commands work and which ones don't work or work with deviations in relation to the described tasks. The implementation of each task is accompanied by necessary modifications and pictures of each newly created code from the Scratch environment. All tasks from the textbook Robotics with Lego Mindstorms for the 2nd grade of primary school can be completed in Scratch with some modifications.

Key words: robotics, LEGO Mindstorms, Scratch, textbook, upper primary schools.

ROZŠÍŘENÍ SOFTWARU SCRATCH PRO ROBOTICKOU UČEBNÍ POMŮCKU LEGO MINDSTORMS

Abstrakt: Článek se zabývá testováním rozšíření online blokového programovacího prostředí Scratch pro LEGO Mindstorms EV3 na úlohách z učebnice Robotika s LEGO Mindstorms pro 2. stupeň základní školy vzniklé v rámci projektu PRIM (Podpora rozvíjení informatického myšlení). Učebnice byla sepsána pro SW, který se dnes již nevyužívá. Z těchto důvodů jsme se rozhodli ověřit, zda lze všechny tyto úlohy naprogramovat v blokovém programovacím jazyce Scratch a s pár vybranými čtenáře seznámit.

* Autor pro korespondenci: pklofac@pf.jcu.cz

Poukazujeme na fakta, jak moc je rozšíření Scratche kompatibilní se stavebnicí, které příkazy fungují a které naopak nefungují či pracují s odchylkami ve vztahu k popisovaným úlohám. Realizace jednotlivých úloh je doplněna o nutné úpravy a obrázky jednotlivých nově vytvořených kódů z prostředí Scratch. Všechny úlohy z učebnice Robotika s LEGO Mindstorms pro 2. stupeň ZŠ lze za určitých úprav ve Scratchi splnit.

Klíčová slova: robotika, LEGO Mindstorms, Scratch, učebnice, 2. stupně ZŠ.

1 Úvod

V lednu 2021 zveřejnilo ministerstvo školství nový RVP pro základní vzdělávání (MŠMT, 2021). Tento dokument definuje národní kurikulum, jehož součástí je i nové pojetí informatiky. Nově se začaly používat termíny, které vyzdvihují cíle vzdělávání a rozvoj jedince: informatické myšlení a digitální gramotnost. Nová informatika se bude nově věnovat především programování, algoritmizaci a robotice (Vaníček, 2021).

Zmiňované informatické myšlení je stále více považováno za nedílnou součást moderní školy. Kabátová a kol. uvádí požadavky ze stran pedagogů, tvůrců vzdělávací politiky i ze stran vědců, aby se informatické myšlení stalo součástí základního vzdělávání všech dětí (Kabátová a kol., 2016). Autoři však uvádí různé důvody k tomuto smýšlení. Část považuje informatické myšlení za novou gramotnost, kterou by si měl každý rozvíjet z důvodu, aby se stal plně rozvinutým jedincem v digitálním světě. Jiní se domnívají, že by žáci měli rozvíjet své informatické myšlení, aby jej mohli využít v pozdějším pracovním zaměření v oblasti informatiky (Kabátová a kol., 2016; Kafai, 2016).

Programování a robotika představují možnou cestu např. v užití jazyků, kterými se žáci domluví s počítači i roboty. Robotika skýtá opravdu rozmanitou škálu možností. Beze sporu je robot pro žáky 2. stupně motivačním prvkem, může být ovšem také pomůckou pro rozvoj informatického myšlení (Kupilíková, Simbartl, 2016). Edukační robotika si nachází již dobré desetiletí své místo v kvalitním a moderním vzdělávání. Její implementace se pomalu přesouvá do stále nižších stupňů vzdělání. Roboti se stávají nezbytnou součástí našeho života, své uplatnění naleznou v různých oborech i domácnostech (Miková a kol., 2021). Úkolem pedagogů je bezpochyby vzdělávání a výchova, ale také příprava žáků na budoucí profesi. Robotické pomůcky nabízejí příležitost k praktickému pochopení věcí, se kterými se žáci setkávají v každodenním životě, ale kterým plně nerozumí, jako jsou typy senzorů, odůvodňování poruch (softwarové chyby) a problémů

s připojením (odpojení Wi-Fi, Bluetooth). Tyto pomůcky můžeme dále rozdělit na robotické hračky a robotické stavebnice.

Robotické stavebnice se od robotických hraček odlišují svojí flexibilitou. Autoři Ben-Ari a Mondada zmiňují, že můžeme navrhnout a postavit robota, který bude plnit konkrétní úkol, omezeni jsme pouze svou představivostí (Ben-Ari, Mondada, 2018). Robotické stavebnice podporují kreativitu, zručnost a týmovou spolupráci, která je jednou z klíčových kompetencí k uplatnění na pracovním trhu. Robotické stavebnice poskytují spojení technologií, inženýrství, matematiky, umění a vědy v praxi (Chaudhary a kol., 2017; Tengler a kol., 2021). Na základě výsledků výzkumné zprávy Baťka bylo zjištěno, že nejvíce používanou robotickou stavebnicí na území České republiky je LEGO Mindstorms EV3, nebo starší model NXT (Baťko, 2017; LEGO, 2022). Tato výzkumná zpráva dnes zcela jistě není aktuální, ale můžeme se domnívat, že školy zakoupené robotické stavebnice stále využívají.

Robotická stavebnice LEGO Mindstorms EV3 odkazuje na programovací prostředí Robolab, které spočívá v přetahování bloků do horizontálně jdoucího kódu. Avšak tento styl programování je takřka ojedinělý a pro další programování nemá žádné jiné uplatnění. V posledních letech mají žáci 1. a 2. stupně základních škol tu možnost začít programovat ve vizuálním programovacím jazyce Scratch. Z toho vyplývá požadavek při práci s roboty používat programovací prostředí, které svým vzhledem a stylem funkčně a vizuálně připomíná Scratch, což žákům usnadní orientaci v programovém prostředí.

2 Testování úloh

Robotické úlohy, které vznikly v rámci projektu PRIM pro robotickou stavebnici LEGO Mindstorms EV3, jsou sepsány pro práci v zastaralém SW LEGO Mindstorms Education. Naším úkolem bylo všechny tyto úkoly otestovat a následně naprogramovat v alternativním online blokovém programovacím prostředí Scratch. Všechny přepracované úlohy z učebnice jsou vloženy do sdíleného studia Scratch dostupného na: https://scratch.mit.edu/studios/33795681. Prostředí Scratch autoři učebnice označují jako nevhodné: „*Programovací prostředí Scratch v nové verzi 3 podporuje robotické stavebnice Lego včetně verze EV3. Podpora je však velmi omezená a neumožňuje plnohodnotnou práci (např. výpis na displej, oddělené řízení motorů, celou řadu senzorů a ladění)*“ (Jakeš, 2020).

1/2024, Volume 16, Issue 1
ISSN 1803-537X

Zajímalo nás, zda úlohy lze naprogramovat takovým způsobem, aby výsledek byl totožný či minimálně přibližný tomu z původního programovacího prostředí. Zaměřili jsme se na odchylky, které brání v úspěšnosti plnění jednotlivých dílčích úkonů z úloh, jako jsou např. absence bloků pro výpis na displej nebo omezená funkčnost senzorů (barevný senzor vrací pouze hodnoty světelného odrazu od objektu, distanční senzor nevrací naměřené hodnoty v centimetrech, ale v procentech apod.).

3 Automatická závora

Jedna z nejzajímavějších a zároveň poměrně náročných programátorských úloh z online učebnice iMyšlení byla úloha „Automatická závora" ze 7. kapitoly. Řešení úlohy ve Scratchi je dostupné na: https://scratch.mit.edu/projects/891174479. Ke splnění této úlohy bylo nutné využít světelný senzor pro rozlišení barev. V originální aplikaci LEGO® MINDSTORMS® EV3 byl k dispozici dedikovaný blok, který dokázal barvy za příznivých světelných podmínek bezpečně rozpoznat na rozdíl od rozšířeného prostředí Scratche, které podobně dedikovaný blok postrádalo. Ve Scratchi se totiž nacházel pouze blok rozlišující množství světla dopadávajícího na čočku senzoru odrážejícího se od předmětu. Z tohoto důvodu bylo nutné učinit dodatečné kroky při programování této úlohy.

Cíl úlohy Automatická závora
Cílem úlohy je naprogramovat automatickou závoru (mýtnou bránu), která po přiložení jedné ze tří různě barevných karet umožní průjezd auta zdvižením závory. Pokud by se auto pod bránou zdrželo nebo stále projíždělo, brána by ho měla detekovat pomocí ultrazvukového senzoru, auto upozornit zvukovým signálem a závoru po uvolnění místa zavřít. Dále úloha propojuje práci s barevným senzorem a displejem kostky (bricku). Pro každou přiloženou barvu by měl program zobrazit různý text na displeji a upravit čas průjezdu pro odlišný typ vozidla.

Výběr vhodného osvětlení
Vzhledem k faktu, že správné fungování programu se opírá o data ze světelného senzoru, bylo nutné zvolit vhodné osvětlení pro co možná nejlepší funkčnost senzoru. Před samotným testováním vhodného osvětlení jsme museli otestovat schopnost senzoru vnímat odražené světlo. Odrazivost světla jsme opakovaně testovali na bílé podložce a jako zdroj světla nám sloužila LED svítilna umístěná

1/2024, Volume 16, Issue 1
ISSN 1803-537X

v různých vzdálenostech od senzoru. Maximální naměřená hodnota odraženého světla dosahovala hodnoty 86 jednotek ze 100.

Samotné testování vhodného osvětlení bylo rozděleno do tří odlišných místností po deseti opakováních. Každá z místností měla jiné světelné podmínky. Cílem testování bylo zjistit dovednost senzoru diferencovat mezi třemi barvami, jmenovitě červenou, modrou a zelenou. Robot byl ve všech testovacích případech umisťován na podlahu místnosti, abychom co nejvěrněji simulovali žákovskou práci s robotem.

Prostorem pro prvních deset testů byla tmavá místnost s jedním stálým světelným zdrojem. Rozhodli jsme se jej otestovat z důvodu neměnných světelných podmínek. Po dokončení testovací sekvence jsme zjistili, že světelné podmínky byly nedostatečné. Senzor u trojice přiložených barev dosahoval maximální hodnoty 5 jednotek a nebylo možné mezi barvami rozlišovat.

Prostorem druhé sady testů byla místnost s mnoha zdroji světla. V této místnosti jsme po sekvenci testů dosahovali obdobných výsledků v porovnání s prvním prostředím. Všechny hodnoty naměřené senzorem se pohybovaly okolo hodnoty 14 jednotek a stejně jako u první sady testů nebylo možné mezi barvami rozlišovat.

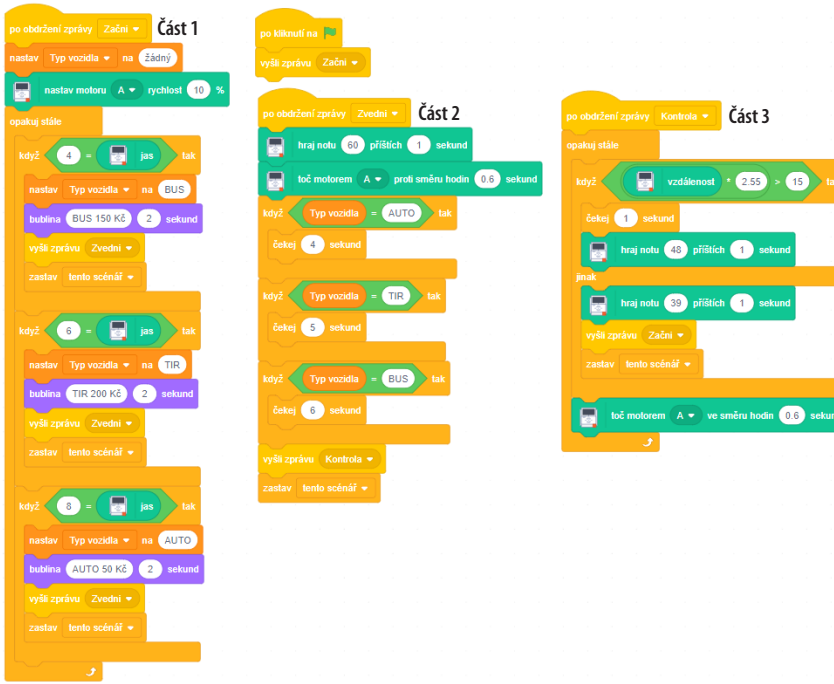
Posledním, třetím prostorem byla místnost s přírodním tlumeným světlem. Po vykonání všech testů jsme zjistili, že pro senzor byly tyto podmínky nejpříznivější, protože pouze v tomto prostředí senzor mohl efektivně pracovat s odlišnými číselnými hodnotami odraženého světla pro každou z barev, jmenovitě červená 8 jednotek, modrá 6 jednotek a zelená 4 jednotky. Nadále jsme pokračovali ve tvorbě blokového kódu za těchto podmínek.

Zmiňovaných podmínek ze třetí sady testů se dalo nejlépe dosáhnout v dopoledních hodinách jarních měsíců od přibližně 8:00 do přibližně 10:30. Jsme si vědomi, že se jedná o určité proměnlivé časové období, nemůžeme tedy tento časový interval nikterak zobecnit. Nicméně v tomto časovém intervalu mělo odražené světlo z bloků stálou hodnotu. Naměřené hodnoty barev jsme získali přikládáním barevných bloků, viz Obrázek 1 přibližně pět milimetrů od senzoru.

Obr. č. 1: Barevné bloky využité k testování světelného senzoru

Programování úlohy

Program byl rozdělen do tří částí, z důvodu snazší orientace viz ilustrační Obrázek 2. Celý program lze spustit zelenou vlaječkou, která je spojena s blokem, který vyšle zprávu „Začni“. Tato zpráva umožní aktivaci první ze tří částí programu.



Obr. č. 2: Blokový kód k úloze Automatická závora

První část programu aktivovaná zprávou „Začni“ se stará o rozlišení barvy přiložené karty. Proměnná „Typ vozidla“ je na začátku iniciována na hodnotu „žádný“. V nekonečné smyčce se neustále kontroluje barva vnímaná barevným senzorem. Uvnitř smyčky se nachází trojice podmínek kontrolujících tři zmiňované barvy. Výpis na displej kostky (bricku) musel být nahrazen výpisem do okna Scratche pomocí bloku bublina (viz Obrázek 3), protože rozšíření Scratche neumožňuje pracovat s displejem kostky (bricku). Každá z podmínek obsahuje blok nastavující proměnnou „Typ vozidla“ na příslušný typ, blok vysílající zprávu „Zvedni“ a velice

1/2024, Volume 16, Issue 1
ISSN 1803-537X

důležitý blok „Zastav tento scénář“, který zastaví smyčku. Smyčka musí být zastavena kvůli možnému opakujícímu se přiložení barevné karty, což by následně vedlo k narušení chodu celého programu.

Obr. č. 3: Ukázka výpisu textu

Druhá část programu aktivovaná zprávou „Zvedni“ obstarává zdvih ramena závory po různě dlouhou dobu pro každý ze tří typů vozidel. Nastavení prodlevy mezi otevřením a zavřením závory je zajištěno podmínkovými bloky, které kontrolují hodnotu proměnné „Typ vozidla“. Před sklopením ramena brány se musí ověřit prostor pod ním. Toto je zajištěno vysláním zprávy „Kontrola“. Ke konci této části je umístěn blok „Zastav tento scénář“ pro dodatečnou bezpečnost.

Ve třetí, finální části programu aktivované zprávou „Kontrola“ je ověřováno volné místo pod ramenem závory. Toto ověření obstarává ultrazvukový senzor, který zjišťuje vzdálenost překážky. V podmínce si můžeme všimnout násobení hodnoty ze senzoru číslem 2,55. Byli jsme nuceni tento mezivýpočet do programu zakomponovat z jednoho prostého důvodu: rozšíření Scratche nepracuje s hodnotami od 0 do 255 centimetrů jako v LEGO aplikaci, ale se škálou od 0 do 100 procent. Pro zjištění reálné vzdálenosti od ultrazvukového senzoru jsme museli použít tento mezikrok. Pokud je pod ramenem prostor uvolněn, rameno se může po zaznění varovného tónu sklopit a vyšle se zpráva „Začni“, která znovu spustí program tak, aby mohla být přiložena další barevná karta. Po vyslání zprávy se musí daný scénář ukončit pomocí bloku „Zastav tento scénář“, aby nedocházelo ke zbytečné detekci překážky senzorem. Pokud pod ramenem není prostor po uplynutí času uvolněn, začne se ozývat varovný zvuk pro neprodlené uvolnění prostoru. Obě podmínky jsou zakomponovány v nekonečné smyčce, která obstarává neustálou kontrolu prostoru pod ramenem závory.

1/2024, Volume 16, Issue 1
ISSN 1803-537X

Porovnání blokových programovacích prostředí
Rozšíření Scratche neumožňuje využívat plný potenciál stavebnice, jako je tomu u aplikace od LEGA. Je nutné převádět jednotky vzdálenosti z procent na centimetry pomocí mezivýpočtu. Senzor pro snímání barvy funguje pouze jako senzor na odrážené světlo od předmětu za specifických podmínek a není schopen dosáhnout maximální možné hodnoty 100 procent. Motory se nedokáží pohybovat do doby, než je podmínka zastaví, ale pohybují se pouze po dobu sekund. Pozitivně vnímáme možnost volání různých úseků/částí kódu pomocí bloku „Vyšli zprávu". Tyto bloky usnadňují řešení úloh oproti řešení v aplikaci. Můžeme konstatovat, že s výše uvedenými úpravami lze úlohu plnohodnotně splnit.

4 Adaptivní tempomat

Zajímavým programátorským oříškem byla úloha z 8. kapitoly učebnice, která kombinovala využití ultrazvukového senzoru a motorů. Robot by měl zvládat měnit rychlost a směr pohybu na základě přijímaných dat z ultrazvukového senzoru. Řešení úlohy ve Scratchi je dostupné na: https://scratch.mit.edu/projects/890302037.

Cíl úlohy Adaptivní tempomat
Cílem úlohy bylo naprogramovat plně automatizovaného pojízdného robota, který pomocí naměřené vzdálenosti od překážky bude neustále měnit svou rychlost do doby, než se před překážkou bude nacházet ve vzdálenosti 20 centimetrů. Pokud by nastal pohyb překážky vpřed či vzad, robot musí být schopen reagovat na tyto změny simultánním pohybem vpřed či vzad. Na displeji kostky (bricku) mají být zobrazována data o vzdálenosti a rychlosti. Při pohybu vzad bude robot vydávat zvuk.

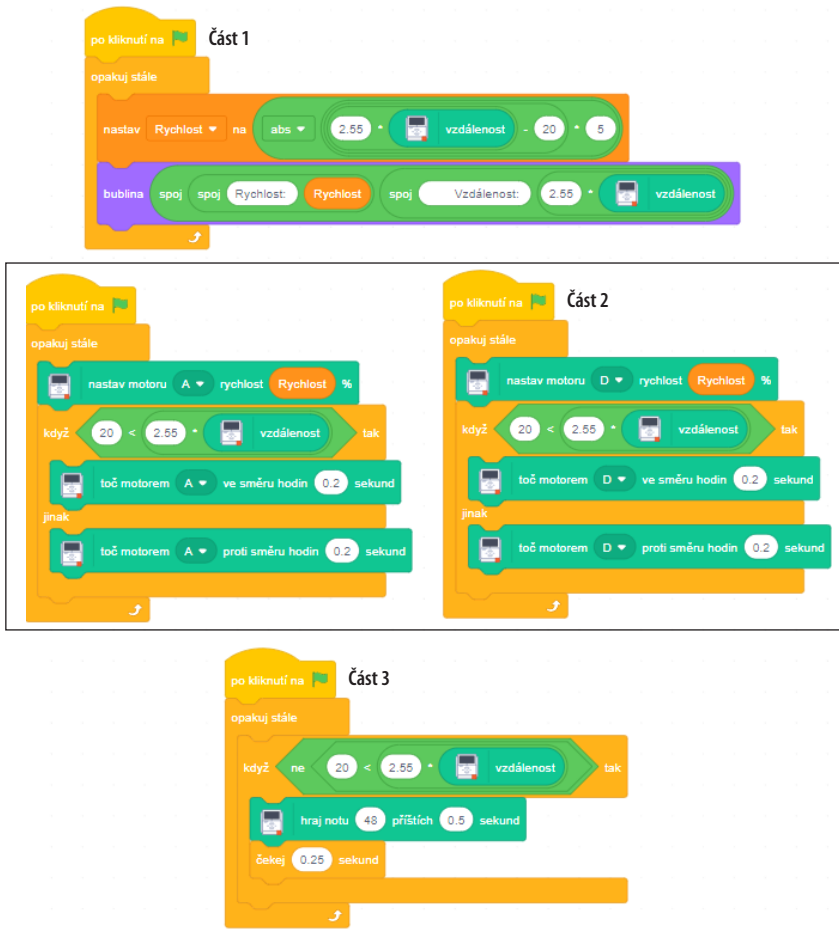
Pohyb robota
Rozšířené prostředí Scratche neumožňuje pohybovat motorem do doby, než bude motor podmínkou zastaven, z těchto důvodů jsme byli nuceni používat jediný blok pro pohyb, který umožňoval pohybovat motorem pouze po dobu časového intervalu.

Několika pokusy jsme zjistili, že ideálním časovým intervalem je 0,2 sekundy. Pokud byla použita hodnota menší než 0,2 sekundy, tak byl pohyb velice trhavý.

Naopak za použití většího časového intervalu se detekce překážky stala pomalou a robot nestíhal přiměřeně reagovat na přibližující se či vzdalující se překážku.

Programování úlohy

Celý program je opět rozdělen do tří částí (viz ilustrační Obrázek 4). Program musí být spouštěn přes zelenou vlaječku, protože je nutné všechny jeho části spustit v tentýž moment.



Obr. č. 4: Blokový kód k úloze Adaptivní tempomat

1/2024, Volume 16, Issue 1
ISSN 1803-537X

První část programu neustále mění rychlost v závislosti na vzdálenosti robota od překážky. Opět jsme byli nuceni používat mezivýpočet pro reálnou vzdálenost senzoru od překážky. Na rozdíl od originální aplikace je zde hodnota rychlosti v absolutní hodnotě, rozšíření Scratche nedokáže pracovat se zápornou hodnotou rychlosti. Dále tato část programu obsahuje blok „Bublina", který do okna Scratche vypisuje aktuální vzdálenost od překážky a rychlost. Vše je zakomponováno do nekonečné smyčky z důvodu neustálé okamžité reakce robota na nastalé změny.

Druhá část programu se skládá z částí pro pohyb motoru A a D. Rozšíření Scratche neumožňuje pohybovat motory souběžně pomocí jednoho bloku, je tedy nutné motory naprogramovat odděleně. V každé části se nachází blok nastavující rychlost motoru a podmínka, která rozhoduje o pohybu vpřed či vzad. Takto naprogramované motory dokáží pracovat souběžně.

Třetí část programu se stará o vydávání varovného zvuku při couvání. Za blokem tónu se nachází blok „Čekej", aby zvuk nebyl nepřetržitý.

Porovnání blokových programovacích prostředí

Jedná se o totožný případ jako v předešlé úloze, a sice nutnost provést mezivýpočet pro převod změřené vzdálenosti senzorem do reálné vzdálenosti. Plynulého pohybu motorů nelze docílit a jediný možný pohyb je pohyb trhaný. Dále rozšíření nedokáže pracovat se zápornými hodnotami rychlosti, jako tomu je v aplikaci LEGO Mindstorms Education. Zpětný chod robota tak musí být řešen dodatečným přidáním podmínky. S jistými nedokonalostmi, jako např. zmiňovaný trhaný pohyb, je úlohu možné plnohodnotně splnit.

5 Závěr

LEGO Mindstorms jako jedna z velice oblíbených robotických stavebnic v procesu vzdělávání má hojné zastoupení různých programových – ať už online nebo softwarových – prostředí. Právě jednou z variant je prostředí Scratch. Po kompletním přepracování všech úloh z učebnice Robotika s LEGO Mindstorms pro 2. stupeň ZŠ do rozšířeného prostředí Scratch můžeme tvrdit, že všechny vytvořené úlohy lze ve Scratchi splnit s jistými obtížemi, kterými jsou: nemožnost ovládání motorů po dobu časového intervalu či pomocí otáček, nemožnost práce s displejem, barevný senzor neschopný rozeznávat barvu, malý repertoár zvuků a další. Domníváme se, že prostředí Scratch nebude vhodnou variantou pro

vyučující informatiky, kteří s programováním robotické LEGO stavebnice začínají. Rozšířené prostředí Scratch bychom doporučili učitelům informatiky, kteří rádi bádají nad problematikou vzniklých úloh z projektu PRIM a mohli by své znalosti a zkušenosti předávat např. v robotickém kroužku, kde tyto vzniklé úlohy mohou dávat zajímavý podnět, jak je splnit, pokud dané prostředí nenabízí vše potřebné.

Jsme si vědomi absence ověření námi předělaných úloh na žákovském kolektivu. Ověřování úloh žáky nebylo našim cílem z důvodu nevědomosti, zda úlohy v rozšířeném prostředí Scratch lze naprogramovat. Nicméně v následujícím školním roce, v rámci navazujícího výzkumu, budou úlohy otestovány na několika třídách 2. stupně ZŠ.

6 Literatura

Baťko, J. (2017). *Robotika ve výuce na základních školách v České republice: Výzkumná zpráva.* Dostupné z: https://www.kvd.zcu.cz/cz/dokumenty/Batko_robotika_ve_vyuce_na_ZS_v_CR.pdf

Ben-Ari, M., & Mondada, F. (2018). *Elements of Robotics*. Switzerland: Springer Cham. Dostupné z: https://link.springer.com/book/10.1007/978-3-319-62533-1

Chaudhary, V., Agrawal, V., Sureka, P., & Sureka, A. (2017). *An experience report on teaching programming and computational thinking to elementary level children using lego robotics education kit.* In: Proceedings – IEEE 8th International Conference on Technology for Education, T4E, pp. 38–41

Jakeš, T., Baťko, J., & Simbartl, P. (2020). *Robotika s LEGO Mindstorms pro 2. stupeň základní školy.* Imysleni.cz. Západočeská univerzita v Plzni, 2020. Dostupné z: https://imysleni.cz/ucebnice/robotika-na-2-stupni-zakladni-skoly-s-lego-mindstorms

Kabátová, M., Kalaš, I., & Tomcsanyiová, M. (2016). *Programming in Slovak Primary Schools.* In: Olympiads in Informatics, 10, s. 125–159. Dostupné z: https://ioi.te.lv/oi/pdf/v10_2016_125_159.pdf

Kafai, Y. B. (2016). *From computational thinking to computational participation in K-12 education.* Communications of the ACM, 59(8), 26–27.

Kuplíková, M., & Simbartl, P. (2016). *Využití robotiky ve výuce na základní škole.* Edukacja – Technika – Informatyka, 16(2), 121–127. DOI: 10.15584/eti.2016.2.15. Dostupné z: http://repozytorium.ur.edu.pl/handle/item/1986

LEGO. (2022). *Lego Mindstorms EV3*. Lego. Dostupné z: https://ruzovka.cz/cs/2-stupen-zs-vxs/16660-lego-education-45544-ev3-zakladni-souprava-vbt.html?gclid=EAIaIQobChMIpt7v4YmL_gIVAszVCh2CwQG3EAAYASAAEgLYXPD_BwE

Miková, K., Budinská, L., & Stenová, B. (2021). *Analýza edukačných robotických hračiek dostupných na Slovensku.* Banská Bystrica. Dostupné z: http://www.didinfo.net/images/DidInfo/files/DIDINFO_2021_zbornik.pdf

MŠMT (2021). *Opatření ministra školství, mládeže a tělovýchovy, kterým se mění Rámcový vzdělávací program pro základní vzdělávání* (Frame educational programme for basic education). MŠMT, Praha. https://www.msmt.cz/file/54860_1_1

Tengler, K., Kastner – Hauler, O., & Sabitzer, B. (2021). „*Identifying Preliminary Design Principles for a Robotics-based Learning Environment*," 16th International Conference on Computer Science & Education (ICCSE), pp. 771-776, doi: 10.1109/ICCSE51940.2021.9569250.

Vaníček, J. (2021). *Towards a compulsory computing curriculum at primary and lower-secondary schools: the case of Czechia.* In: Barendsen, E., Chytas, Ch. (eds): Informatics in Schools. Rethinking Computing Education. ISSEP. Lecture Notes in Computer Science, vol. 13057. Cham: Springer, p. 109–120, 2021. https://doi.org/10.1007/978-3-030-90228-5_9