



Pedagogická
fakulta
Faculty
of Education

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta
Katedra informatiky

Diplomová práce

Výuka algoritmizace v EasyLogo na 1. stupni ZŠ

Vypracovala: Martina Bartíková
Vedoucí práce: doc. PaedDr. Jiří Vaníček, Ph.D.

České Budějovice 2017

Název práce: Výuka algoritmizace v EasyLogo na 1. stupni základní školy

Autorka: Martina Bartíková

Katedra: Katedra informatiky

Vedoucí práce: doc. PaedDr. Jiří Vaníček, Ph. D.

E-mail vedoucího: vanicek@pf.jcu.cz

ANOTACE

Práce se zabývá výukou programování na prvním stupni základních škol ve výukové aplikaci EasyLogo. Teoretická část obsahuje analýzu výuky původní sady krátkých programovacích úloh. Na základě pozorování výuky ve školách a následné interakce s učiteli a žáky byla tato sada upravena a doplněna o nové úlohy.

Nová výuková sada obsahuje 52 úloh rozdělených do sedmi oddílů, které dodržují postup výuky na základě vyučovaných kroků. Součástí práce jsou popisy a náhledy nově vytvořených úloh, metodické pokyny pro učitele a jako příloha je uveden kompletní seznam úloh rozdělený do oddílů.

KLÍČOVÁ SLOVA

Programování, algoritmizace, EasyLogo, 1. stupeň základní školy, výuka

Title: Education of algorithmization in EasyLogo on first grade of primary school

Author: Martina Bartíková

Department: Katedra Informatiky

Supervisit: doc. PaedDr. Jiří Vaniček, Ph. D.

Supervisor's e-mail address: vanicek@pf.jcu.cz

ANOTATION:

This master thesis deals with programming education on first grade of primary schools in application EasyLogo. The theoretical part consists of analysis of education of an original set of short programming exercises. Based on observation of teaching at schools and following interactions with teachers and students was the set modified and supplemented by new exercises.

The new educational set includes 52 exercises divided into seven sections which keep the teaching process following the taught steps. The master thesis consists of descriptions and previews of new exercises, guidelines for teachers and as an attachment there is a complete list of exercises divided into sections.

KEYWORDS:

Programming, algorithmization, EasyLogo, first grade of primary school, education

Prohlašuji, že svoji diplomovou práci jsem vypracovala samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě - v úpravě vzniklé vypuštěním vyznačených částí archivovaných Pedagogickou fakultou elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

24. 4. 2017

Poděkování

Za odbornou pomoc při zpracování diplomové práce bych chtěla na tomto místě poděkovat vedoucímu práce panu doc. PaedDr. Jiřímu Vaníčkovi, Ph. D., který mi v průběhu psaní dával cenné rady. Dále děkuji za podporu výzkumu grantovému projektu GAJU 121/2016/S.

ZADÁNÍ DIPLOMOVÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Martina BARTÍKOVÁ**
Osobní číslo: **P12171**
Studijní program: **M7503 Učitelství pro základní školy**
Studijní obor: **Učitelství pro 1. stupeň základních škol**
Název tématu: **Výuka algoritmizace v EasyLogo na 1. stupni ZŠ**
Zadávací katedra: **Katedra informatiky**

Z á s a d y p r o v y p r a c o v á n í :

Diplomová práce bude zaměřena na výuku základů algoritmizace na 1. stupni ZŠ, na používání programovacího prostředí EasyLogo. EasyLogo je jednoduché free prostředí na principu Logo a želví grafiky, speciálně vytvořené pro výuku malých dětí v programování. Programem, který vytvoří žák, je ovládnutí objekt, který se pohybuje po obrázkovém pozadí a kreslí za sebou čáru.

Diplomantka nejprve ověří stávající sadu programovacích úloh ve výuce (např. rozhovory s učitelkami, které ji vyučovaly, pozorováním při výuce) a poté tuto sadu úloh pro začátečníky inovuje. Bude sledovat cíl vytvořit sled krátkých programovacích etud pro výuku porozumění jednotlivým programovacím strukturám (kladení příkazů za sebou, opakování, podprogram). Sama vymyslí úlohy a implementuje jejich zadání do prostředí EasyLogo včetně grafiky a vhodně jimi doplní stávající sadu.

Diplomantka inovovanou sadu úloh ověří ve výuce (své nebo ve spolupráci s učitelkami) ve 4. či 5. třídě ZŠ. Ověřování bude probíhat na 2 (3) základních školách, v hodinách informatiky. Na základě připomínek a poznatků z ověření diplomantka sadu úloh následně bude upravovat. Výuková sada bude finálně obsahovat kromě úloh i metodické poznámky k úlohám a stylu práce učitele.

Diplomová práce bude obsahovat teoretickou a empirickou část. Teoretická část bude zaměřena na výuku programování pro malé děti a na program EasyLogo, jeho stávající výukovou řadu, poznatky a komentáře k jednotlivým úlohám. Empirická část bude obsahovat popis nově vytvořených úloh, spolu s metodickými poznámkami a analýzou ověření výuky, komentáři k výkonům žáků, komentáři k jednotlivým úlohám a případné návrhy úprav a dalších vylepšení sady úloh pro výuku programování pro tuto věkovou kategorii.

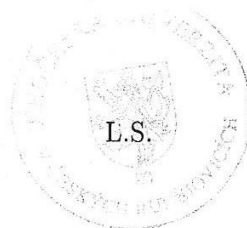
Rozsah grafických prací: **CD ROM**
Rozsah pracovní zprávy: **60**
Forma zpracování diplomové práce: **tištěná**
Seznam odborné literatury: **viz příloha**

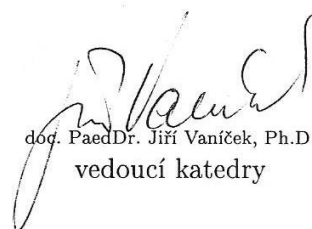
Vedoucí diplomové práce: **doc. PaedDr. Jiří Vaníček, Ph.D.**
Katedra informatiky

Datum zadání diplomové práce: **18. listopadu 2015**
Termín odevzdání diplomové práce: **30. dubna 2017**



Mgr. Michal Vančura, Ph.D.
děkan




doc. PaedDr. Jiří Vaníček, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 18. listopadu 2015

Obsah

Seznam zkratk	11
1. Úvod	12
1.1 Cíle práce	13
1.2 Metody práce	14
2. Výuka programování na ZŠ a přehled vybraných prostředí	17
2.1 EasyLogo	17
2.1.1 Autorský program GEN pro tvorbu úloh v EasyLogo	19
2.2 Scratch	20
2.3 Baltík	20
2.4 Imagine Logo	20
3. Materiály pro výuku informatiky na 1. stupni ZŠ	21
3.1 Informatika pro 1. stupeň základní školy	21
3.2 Programování pro děti: naučte se programovat při tvorbě skvělých her	21
3.3 Baltík: Učebnice programování nejen pro děti	21
4. Původní sada úloh EasyLogo	22
4.1 Mód přímý	22
4.2 Tvorba programu	23
4.3 Opakování	24
4.4 Řešíme problémy	26
4.5 Opravujeme chyby	26
4.6 Procedury	27
5. Prvotní pozorování postupu výuky v EasyLogo na ZŠ	29
5.1 Pozorování výuky programování na ZŠ v prostředí EasyLogo	29
5.2 Reakce žáků	30
5.3 Žákovské chyby při řešení úloh	31
5.4 Závěr z prvotního pozorování výuky EL	35
6. Tvorba nové sady úloh	38
6.1 Tvorba nových úloh pro aplikaci EasyLogo	38
6.1.1 Ruční tvorba úloh	39
6.1.2 Tvorba úloh pomocí autorského programu GEN	42
6.2 Upravené původní úlohy	44
6.3 Přehled nově vytvořených úloh	49
7. Ověření nové sady ve výuce	57
7.1 Ověření ve výuce na základních školách	58

7.2	Ověření učiteli.....	58
7.3	Metodika výuky.....	59
7.3.1	Osnova výuky	59
8.	Závěr.....	63
	Literatura	65
	Publikace	67
	Přílohy	68

Seznam zkratek

EL – EasyLogo

NÚV – Národní ústav pro vzdělávání

RVP – Rámcový vzdělávací program

1. Úvod

Pro svou diplomovou práci jsem si zvolila téma algoritmizace v EasyLogo na 1. stupni ZŠ hlavně proto, že jsem chtěla žákům na prvním stupni základních škol hravou formou poskytnout výuku programování. Samotná výuková aplikace EasyLogo (dále jen EL) mě zaujala už v nižších ročnících mého studia na VŠ, její jednoduchost, okamžitá vidina výsledku svého jednání a její dostupnost.

Ze své praxe jsem pochytila, že v dnešní době se učitelé výuce programování věnují zřídka kdy. Důvodem je hlavně neznalost aplikací a někdy i neochota se do složitějších z nich pouštět a hlouběji se jim věnovat. Aplikací pro výuku na základních školách, potažmo na pro 1. stupeň ZŠ, ani mnoho není, aby si učitelé mohli vybírat ze širší škály.

Moderní doba klade čím dál tím větší nároky na člověka, co se týče digitálních technologií. V souvislosti s tím se jedná i o rozvoj inforatického myšlení. Instituce International Society for Teachnology in Education (ISTE, 2011) definuje inforatické myšlení, neboli „Computational Thinking“, jako proces, který je postavený na snaze řešit problémy a který musí vykazovat určité znaky. Mezi takové znaky patří např. řešení realizované formou algoritmu (naplánované kroky) nebo hledání, analyzování a implementace možných řešení s cílem dospět k co možná nejúčinnějšímu výsledku. (ISTE, 2011)

V rámci své diplomové práce jsem měla možnost, díky grantu GAJU navštívit celostátní konferenci Počítač ve škole v Novém Městě na Moravě. Zde se hovořilo i o rozvoji digitální gramotnosti a inforatického myšlení a to v rámci revize rámcového vzdělávacího programu (RVP), plánované a připravované Národním ústavem pro vzdělávání (NÚV). V této revizi se mimo jiné začleňuje do předmětu inforaticky výuka algoritmizace a programování. Inforatické dovednosti by se měly začleňovat do ostatních předmětů, např. v českém jazyce to bude práce s textovým editorem, v matematice práce s daty apod. a měly by se vyučovat už od prvních ročníků základních škol, kde žáci budou požívat třeba tabletová zařízení, pomocí nichž budou skládat obrázek a správně si jej uložit, v dalším ročníku už budou psát jednoduché texty atd. Se změnou RVP bude potřeba nových výukových materiálů pro učitele. Mou prací vznikl výukový materiál pro učitele prvního stupně základní školy, aby mohly do svých hodin inforaticky zařadit právě výuku algoritmizace

a programování prostřednictvím aplikace EasyLogo s upravenou a rozšířenou sadou úloh.

Myslím, že aplikace EL je jednoduchá na pochopení, učitel může okamžitě zkontrolovat správnost řešení, popřípadě navádět žáky k logickému uvažování a k nacházení dalších typů řešení úloh.

Jednoduchá je i dostupnost aplikace. EL je k dispozici zdarma ke stažení na serveru „knihy.cpress.cz“, jako příloha učebnice Informatika pro 1. první stupeň základní školy (Vaníček, 2012). Soubor, který se musí po stažení pouze rozbalit, je okamžitě připraven k použití a nemusí se provádět žádná další složitá instalace. V mé vlastní výuce tak mohli žáci sami soubor stáhnout do počítače a pouze pomocí instrukcí jej zprovoznit.

Ve své práci představím novou sadu úloh spolu se závěrem z pozorování z hodin informatiky. Původní sadu úloh EL jsem nejprve vyzkoušela v pátých ročnících, všimla si nedostatků, kde pro žáky byl příliš rychlý postup při výuce, kam by se daly zařadit nové úlohy, u kterých úloh by bylo potřeba změnit zadání, a následně jsem celou sadu poupravila. Vznikla tak nová sada úloh, která obsahuje jak úlohy původní, tak s pozměněným zadáním, poupravěné celkově či zcela nové úkoly. Jednotlivé úlohy jsou seřazeny tak, aby byla dodržena posloupnost výuky jednotlivých programovacích kroků a postupovalo se od nejjednodušších úloh k těm složitějším. Takto upravenou sadu jsem znovu vyzkoušela v pátých ročnících a napsala jsem o tom poznatky z hodin.

1.1 Cíle práce

Cílem mé diplomové práce bylo upravit a rozšířit stávající sadu úloh aplikace EL a vhodně do něj zařadit nové úlohy, aby byl dodržen postup výuky programování. V prvním kroku jsem úlohy rozdělila do několika bloků podle využití daných funkcí. Přeházela jsem pořadí stávajících úloh a v některých z nich jsem opravila chyby a doplnila nedostatky. Dále jsem tuto řadu rozšířila o své vlastní úlohy, které jsem vymýšlela na základě poznatků z pozorování výuky. Úlohy byly přidány zejména v místech, kde žáci potřebovali docvičit daný krok.

Takto upravenou sadu úloh jsem otestovala ve výuce informatiky na 1. stupni základních škol, stejně tak jako původní sadu úloh. Poskytla jsem ji opět učitelům,

kteří již EL znali z dřívějších dob a dále i učitelům kteří se s programem setkali poprvé. Testování, ať už prvotní či finální, probíhalo na základních školách v okolí Českých Budějovic. Nová sada byla samozřejmě testována na žácích, kteří neměli žádné zkušenosti s tímto programem, tudíž neznali jeho předchozí verzi.

1.2 Metody práce

Ve své diplomové práci jsem použila několik metod práce, např. ověření – pozorování ve výuce informatiky, rozhovory s učiteli a se žáky, studium podkladů, testování nové sady úloh a v neposlední řadě i testování nového prostředí pro usnadnění tvorby nových úloh pro program EL.

Průzkum ve výuce

Na vybraných školách v okolí Českých Budějovic (ZŠ Grünwaldova, ZŠ Nedabyle) jsem po domluvě s učiteli vyzkoušela původní sadu úloh. Hodiny jsem sama odučila a pozorovala žáky při práci s programem. Žáci dostali základní informace o programu a poté jsme společně postupovali k těžším úlohám. Své poznatky jsem si zapisovala.

Srovnávání výsledků z jednotlivých škol a vyhodnocení

Zápisy z jednotlivých hodin a škol jsem porovnávala a zjišťovala, kde se vyskytují globální problémy v algoritmizaci. Na základě tohoto porovnávání jsem mohla vymýšlet nové úlohy a poupravit ty stávající.

Model ADDIE

Jedná se o model, který umožňuje dívat se na proces návrhu výuky jako na celek. Model obsahuje pět základních fází, které na sebe navzájem navazují. Starková (Starková, 2012) ve své práci uvádí, že je problematické najít přesnou definici modelu ADDIE. Zkratka ADDIE v sobě ukrývá pět slov nazývajících právě základní fáze – Analysis (analýza), Design (návrh), Development (rozvoj), Implementation (uskutečnění), Evaluation (hodnocení). Tyto fáze lze použít při tvorbě nových výukových materiálů či aktivit jako je právě nová sada programu EL.

Fáze analýzy je podle Starkové (Starková, 2012) považována za fázi, kterou je nutno nejvíce promyslet. Sem by bylo možné zařadit pozorování výuky na základních školách, seznámení se žáky a ujasnění jejich možností. V této fázi jsem analyzovala stávající stav výukové řady, potřebnost změn a absenci rozšiřujících úloh.

Do následující fáze mohu zařadit vymýšlení a návrhy nových úloh. Promýšlela jsem obsah nové výukové sady, aby byl vhodný pro žáky prvního stupně.

Fáze rozvoje vytváří konkrétní produkt, tudíž v mé práci to znamená vlastní tvorba úloh do nové sady.

V posledních dvou fázích, tj. uskutečnění a hodnocení, jde hlavně o uskutečnění projektu a o zpětnou vazbu k němu. V mé práci se tedy jedná o ověření ve výuce na prvním stupni a následně o rozhovory s učiteli i se samotnými žáky, kteří aplikaci EL testovali.

Tímto modelem jsem se inspirovala a snažila se dodržet všechny popsání kroky, abych mohla mít dobrý přehled o své dosavadní práci. Zejména i proto, že jsem vytvářela jakýsi „návrh“ výuky prostřednictvím nových úloh zasazených do původní sady. EL, jak už bylo řečeno, obsahuje sadu úloh, které jsou v souvislosti poskládané za sebe, tudíž se podle této aplikace může řídit výuka.

Úprava nové sady

Na základě poznatků z pozorování jsem poupravila zadání jednotlivých stávajících úloh, u některých bylo třeba změnit obrázky na pozadí, aby se dala úloha snadněji hodnotit, a v případě potřeby jsem vytvořila nové vlastní úlohy a vhodně je zařadila do postupu výuky. K nové sadě jsem rovněž vytvořila seznam výukových aktivit pro přehlednost práce s výukovou aplikací.

Do nové sady bylo přidáno 9 zcela nových úloh, pro které jsem vytvořila nové pozadí a vymyslela jsem zadání, a 3 poupravené úlohy, ve kterých bylo upraveno pozadí či zadání.

Ověření

Novou sadu úloh jsem poskytla učitelům základních škol, aby jej vyzkoušeli ve výuce a i já sama jsem si ji vyzkoušela ve své praxi. Někteří učitelé již aplikaci EasyLogo znali a věděli, jak s ním pracovat, jednalo se zejména o mé kolegyně z univerzity, které novou sadu zkoušely v souvislých praxích v rámci studia na VŠ. Ostatním učitelům jsem vysvětlila základní funkce programu. Následně na to jsem s těmito učiteli hovořila a probírala vhodnost výukového postupu a zařazení úloh. Zpětnou vazbu jsem získávala i od samotných žáků.

Aplikaci v nové podobě vyzkoušely i studentky JČU, studující učitelství pro 1. stupeň, stejně jako já, v rámci seminářů při práci s PC. I od nich se mi dostalo zpětných vazeb k mým úlohám.

2. Výuka programování na ZŠ a přehled vybraných prostředí

Programování se na prvním stupni základních škol vyučuje opravdu zřídka. Za hlavní důvod tohoto problému považuji to, že výuka programování není zařazena v Rámcovém vzdělávacím programu (MŠMT, 2013). Dále je to pak absence znalostí různých aplikací a jejich metodiky. Pro výuku programování na základní škole, na 1. stupni, totiž není v českých verzích mnoho aplikací a neposledním negativem je i fakt, že se učitelé mnohdy ani nesnaží programování vyučovat, ba naopak se mu vyhýbají. Podporu pro výuku programování nemají ani formou nabídky různých rozšiřujících školení, přednášek či workshopů.

Při mé práci mě velice zaujala odpověď Steva Jobse, kterou ve svém článku pro server pressreader.com zmiňuje Otakar Schön (Schön, 2016), na otázku „Proč se učit programovat?“ totiž odpověděl: „Myslím, že by se všichni měli naučit programovat, protože vás programování naučí přemýšlet.“ Když se nad touto odpovědí zamyslíme, rozhodně na tom něco bude. Mnoho lidí dnes ztrácí ochotu přemýšlet, v době internetu, kdy každou maličkost objeví na webových stránkách, příklad vypočítá na kalkulačce, a i napsaný text za nás opraví textový editor. Proto je programování v útlém věku, tj. na prvním stupni ZŠ, dobrou příležitostí žáky donutit zapojit své myšlení a logické uvažování.

Několika aplikacím pro výuku programování jsou věnovány i následující podkapitoly mé práce.

2.1 EasyLogo

EasyLogo je jednoduchá aplikace, již je věnována tato práce, určená pro výuku programování. Jedná se o jednoduché programovací free prostředí, v němž se objekt, který vykonává příkaz, pohybuje v prostoru s mřížkou. Autor této výukové aplikace Salanci (2010) přímo uvádí, že „*EasyLogo bylo navrženo pro lidi se základními počítačovými dovednostmi, aby bylo programování a řešení problémů tak jednoduché, jak to je pro ně možné.*“

Jedná se o pedagogicky uzavřené prostředí, tzn., že úlohy v této aplikaci jsou již vytvořené. Žáci plní zadání v režimu „Řešení úloh“. Úlohy jsou řazené za sebou tak,

aby se dodržoval postup výuky programování. Jejich cílem je tak naučit žáky programování od jednoduchých funkcí až po ty složitější. Začíná se s úlohami, kdy žák pohybující se objekt ovládá pouze pomocí směrových šipek a seznamuje se tak s prostředím aplikace, dále se pokračuje přes samotnou tvorbu programu, kdy žák musí vybrat příkazy z příkazové lišty a poskládat je ve správném pořadí do prostoru určeného k vytváření programu (obr. 1) za sebe tak, aby splnil zadání.



Obr. 1 – Prostředí aplikace EasyLogo: 1. Přepínání úloh; 2. Zadání úlohy; 3. Prostor pro tvorbu programu; 4. Lišta s příkazy; 5. Pohyblivý objekt.

Následují úlohy, které se zabývají složitějšími příkazy, pracuje se v nich s předem hotovým tvarem, nebo musejí žáci najít a opravit chybu v již hotovém programu. Svou úlohu zde má i učitel, prostředí EL samo nevyhodnocuje správnost řešení, tudíž je tato role právě na učiteli. Učitel musí znát věci, které učí, tudíž musí znát aplikaci EL, pokud ji chce zařadit do své výuky. EL je dostupné pro učitele svou jednoduchostí a snadným ovládním.

Dále se žáci v aplikaci EL mohou seznámit i s režimem „vlastní tvorba“, kde si mohou vytvářet vlastní program bez zadání. Zároveň je tato aplikace připravena pro tvorbu dalších vlastních úloh, které mohou vymýšlet jak učitelé, tak i žáci samotní.

Proces vytváření takové úlohy však může být pro někoho složitým. K usnadnění této práce byl vytvořen autorský program Gen, v němž se dají snadno nové úlohy vytvořit.

2.1.1 Autorský program GEN pro tvorbu úloh v EasyLogo

Velkým přínosem při samotné tvorbě úloh pro aplikaci EL byl zmíněný program GEN, který vytvořil student Jihočeské Univerzity Stanislav Hupka. Pomocí tohoto programu lze vytvořit novou úlohu bez složitého vypisování programu do Poznámkového bloku. V prostředí se napíše zadání úlohy, nahraje se příslušné pozadí, tvar, který se bude pohybovat a musí se vybrat i mód ve kterém se bude úloha řešit, tzn. normální mód (simple), kde se přetahují příkazy z postranní lišty, přímý mód (direct) zde se tvar želvy ovládá pomocí tlačítek pohybu nebo mód s možností vkládání procedur (procedural). Zároveň se v programu navolí i velikost mřížky a startovní souřadnice pohybujícího se tvaru. Při své práci jsem měla možnost toto prostředí vyzkoušet.

Prostředí autorského programu vidíme na obrázku č. 2.



Obr. 2 – Prostředí autorského programu GEN.

Takto připravená úloha se jen potvrdí a program vygeneruje textový soubor, který se musí uložit do složky obsahující jak příslušné pozadí, tak i tvar zmíněný

v naprogramovaném souboru. Po spuštění aplikace EL je pak tato úloha zcela normálně funkční.

2.2 Scratch

Scratch je programovací jazyk, vytvořený skupinou Lifelong Kindergarten Group. Jak uvádí učebnice věnována této aplikaci Programování pro děti (The LEAD project, 2013) jde o prostředí, ve kterém se vytváří interaktivní příběhy, animace, hry, prezentace aj. Funguje na skládání barevných dílků (příkazů), prostřednictvím nichž se vytvoří program. Tyto dílky jsou jako části stavebnice, které se do sebe zasazují. Jazyk Scratch je dostupný on-line, vytvořená díla zde můžeme uložit na internetu a sdílet je s ostatními lidmi.

2.3 Baltík

Programovací jazyk Baltík je určen pro žáky základních a středních škol. V tomto prostředí se programuje pomocí ikon, které nahradily textové příkazy. Prostor pomáhá pochopit základy programování.

Prostředí je dostupné on-line pouze ve zkušební verzi, která nepodporuje některé jeho funkce.

2.4 Imagine Logo

Jedná se o kompletně objektový jazyk řízený událostmi. Nejedná se o jednoduché prostředí jako je třeba EasyLogo. S tímto jazykem pracují žáci na druhém stupni základních škol a hlavně pak na středních školách.

Imagine Logo je dostupné zdarma pouze v demo verzi, pořízení licence je zpoplatněné jak pro domácí využití, tak pro školy.

3. Materiály pro výuku informatiky na 1. stupni ZŠ

Učebnic pro výuku informatiky na prvním stupni je v ČR velmi málo. Jediná učebnice souhrnně se zabývající všemi vyučoványi oblastmi je učebnice Vaníčka (podrobněji v následující podkapitole), která je určena žákům 4. a 5. ročníku základní školy. Jako další příklad materiálů pro výuku programování jsem vybrala učebnice o výše popisovaných programovacích prostředích.

3.1 Informatika pro 1. stupeň základní školy

Podporou výuky programování je již výše zmiňovaná učebnice Informatika pro 1. stupeň základní školy (Vaníček, 2012). Tato učebnice je všestranně zaměřena a je rozdělena na osm kapitol, jejichž řazení pomáhá učiteli při výuce. Kniha pracuje mimo jiné i s aplikací EL, popsané ve čtvrté kapitole věnující se programování. V této kapitole „Programujeme“ je popsáno několik úloh, které jsou doplněny o jiné úkoly, aktivity a tipy pro žákovu lepší pochopení příkazů. Učebnice je doplněna o barevné ilustrace, které zajistí zájem dětí na prvním stupni základních škol.

3.2 Programování pro děti: naučte se programovat při tvorbě skvělých her

Programování pro děti: naučte se programovat při tvorbě skvělých her (The LEAD project, 2012) vydalo nakladatelství Computer Press. V roce 2013 vyšla tato barevná učebnice podporující programovací jazyk Scratch. Kniha je dělena do deseti základních úrovní, zaměřující se na jednotlivé kroky, a do tří bonusových úrovní. Jednotlivé úrovně postupně seznamují žáka s prostředím Scratch a s jeho možnostmi.

3.3 Baltík: Učebnice programování nejen pro děti

Tato učebnice, Baltík: Učebnice programování nejen pro děti (Pecinovský, 2000), se zaměřuje na programovací jazyk Baltík. Je rozdělena do tří částí a ty se pak dále člení na podkapitoly. Podkapitoly jsou detailně zaměřeny na jednotlivé kroky při tvorbě programu, součástí jsou i obrázky, které zachycují popisovanou část.

4. Původní sada úloh EasyLogo

Tato kapitola obsahuje rozbor původní sady výukové aplikace EL, rozdělení jednotlivých úloh podle problémové oblasti a ukázky vybraných úloh z jednotlivých bloků.

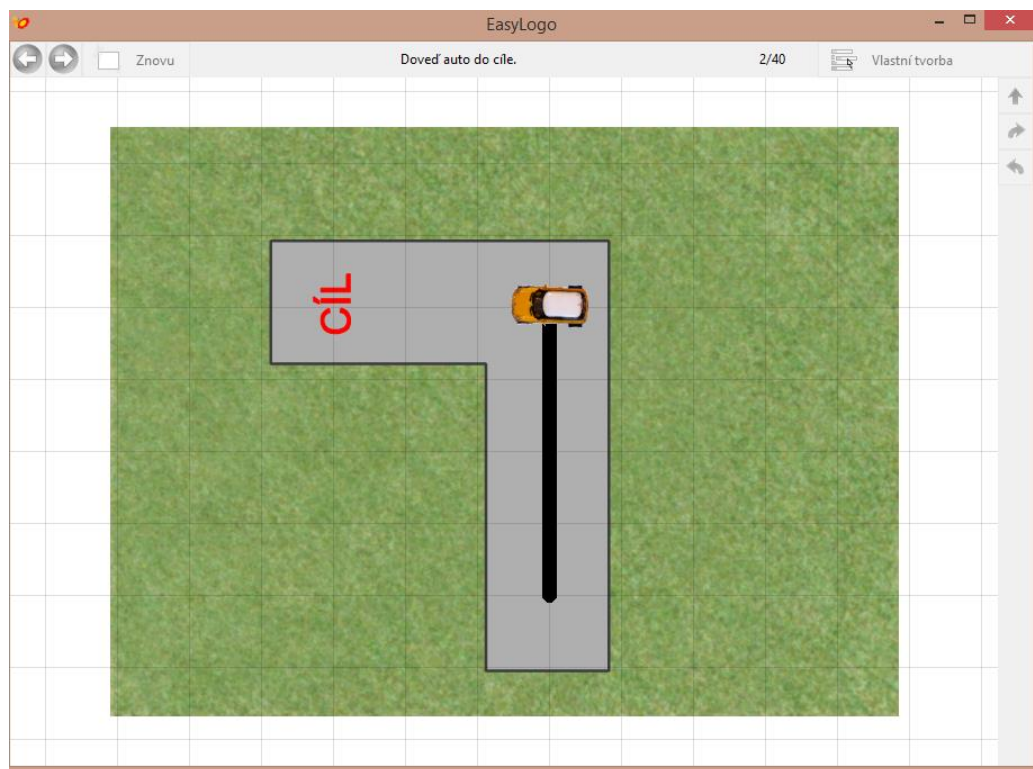
Původní sada úloh v aplikaci EL obsahovala 37 hotových úloh. Jednotlivé úlohy byly vymyšleny tak, aby postupně rozvíjely znalosti programu a rozvíjely dovednosti v oblasti programování. Tyto úlohy byly řazeny za sebou v šesti blocích, které v této práci budu nazývat „oddíly“, na základě vyučovaného kroku – 1. Mód přímý, 2. Tvorba programu, 3. Opakování, 4. Procedury, 5. Opravujeme chyby, 6. Řešíme problémy.

4.1 Mód přímý

V tomto oddílu se využívá k pohybu tvaru želvy pouze směrových šipek. Tvar želvy zde dostal podobu auta, které nesmí vyjet z vyznačené cesty. Jde zde o základní ovládání aplikace, uvědomění si směru, při otočení tvar neudělá krok, ale pouze se otočí o 90° .

V tomto oddílu jsem v rámci upravování sady přidala několik vlastních úloh, aby žáci měli více prostoru na procvičení ovládání a orientace v prostředí.

Pro ukázkou tohoto bloku jsem vybrala hned druhou úlohu (obr. 3). Na obrázku je vidět tato úloha v rozpracované fázi, kde je vidět dráha pohyblivého tvaru, v tomto případě je jím auto. Pro pohyb se používají směrové šipky ze sloupce vpravo.



Obr. 3 – Úloha č. 2; zadání: Doved' auto do cíle. – rozpracovaná úloha.

4.2 Tvorba programu

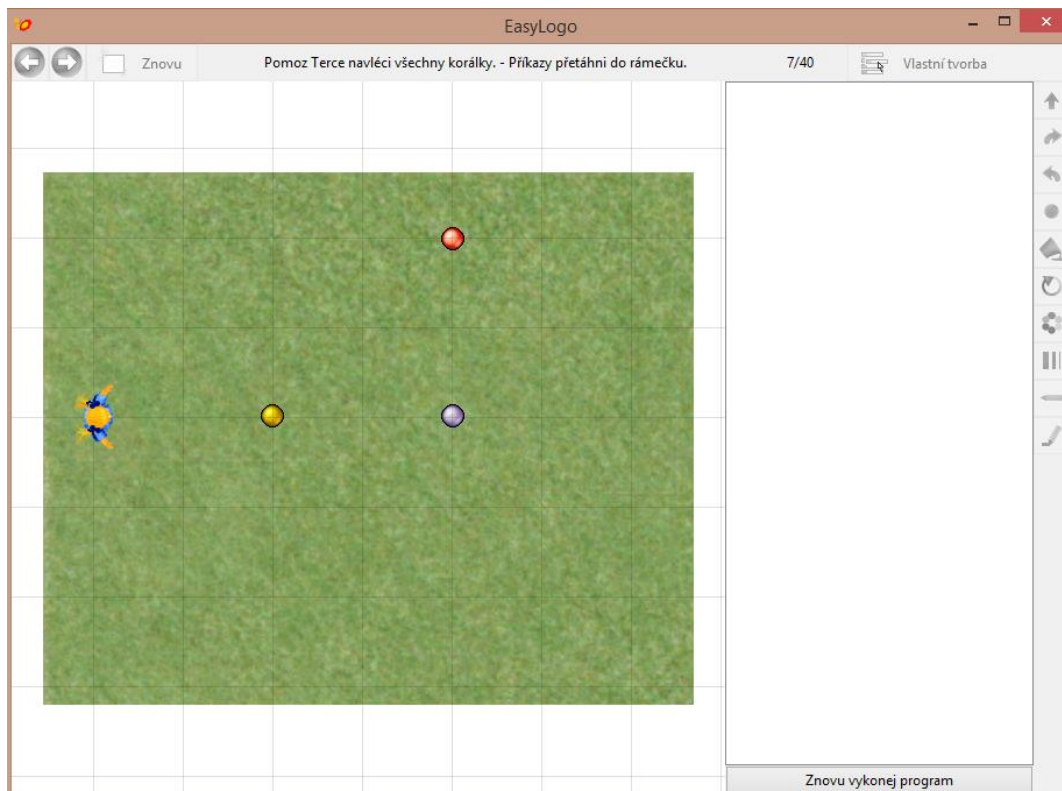
Zde se žáci setkají s přetahováním příkazů na plochu určenou k tvorbě programu. Jedná se o složitější oddíl, už nestačí pouze na příkaz kliknout, ale celý se musí přetáhnout a zařadit na správné místo v programu. Součástí příkazů jsou i čísla, tzn., když použijeme příkaz „dopředu“ a chceme popojít v mřížce o víc než jedno pole rovně za sebou, navolíme u příkazu dopředu číslo např. 3 a želva popojde o tři pole dopředu.

Je nutné si uvědomit, že příkazy mají jít těsně za sebou tak, jak chceme, aby byly provedeny. Výhodou tohoto výukového prostředí je, že ihned po zadání příkazu vidíme pohyb tvaru, tudíž víme, zda provedený krok byl správně či nikoliv.

Zde, v tomto oddílu, jsou opět přidány nově vytvořené úlohy. Jedná se o složitý proces, příkazy se musejí dávat do sloupce na příkazy, a to nejen pouhým kliknutím, nýbrž jejich přetažením na správné místo. Příkazy musejí mít určitý sled, aby byl program vykonán správně, proto je nutné, aby žáci tento oddíl procvičovali po delší čas a pochopili tak jeho systém.

Při výuce tohoto oddílu můžeme použít některé hry, které ve své učebnici informatiky uvádí Vaníček (2012) v kapitole Programování. Můžeme zde třeba vidět naprogramování spolužáka. Žáci vymyslí program, aby jejich kamarád došel např. ke dveřím, ty otevřel, pak zavřel a vrátil se na své místo.

Na obrázku č. 4 vidíme jednu z úloh, které patří do tohoto oddílu. Úkolem je pomoci Terce posbírat všechny korálky, přitom se mají jednotlivé příkazy přetahovat na plochu pro program.



Obr. 4 – Úloha č. 7; zadání: Pomoz Terce navléci všechny korálky. – Příkazy přetáhni do rámečku.

4.3 Opakování

Příkaz „opakuj“ je hlavním bodem tohoto oddílu. Jedná se o složitý krok, nejen proto, že se musí přetáhnout do sloupce pro příkazy a další jednotlivé kroky, které mají být opakované, musejí být jeho součástí tedy vnořeny do samotného příkazu, tudíž je musíme řadit do jakýchsi jeho pod-příkazů, ale zároveň je náročný na pochopení svým principem. Pokud použijeme příkaz „opakuj“, není možné v něm použít další vnořený

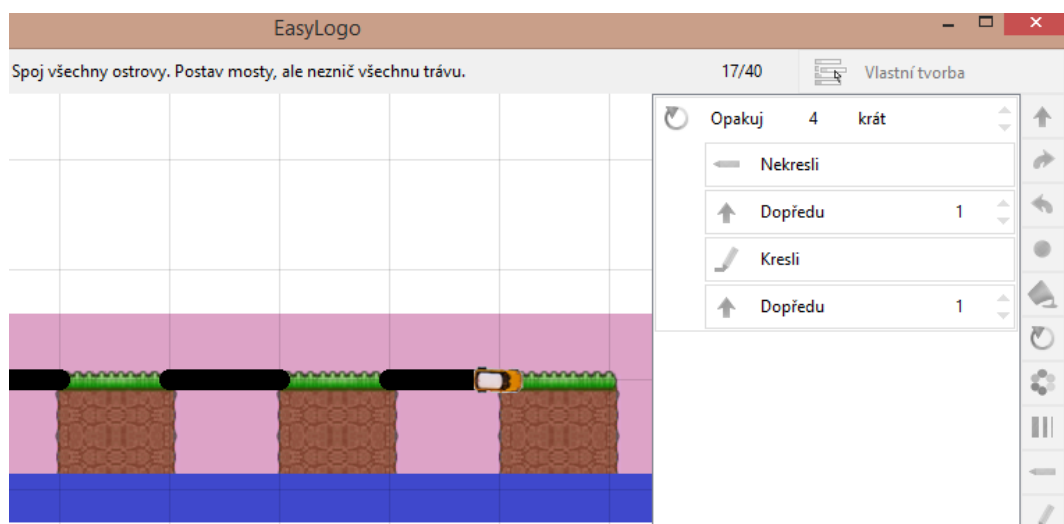
příkaz „opakuj“, ale můžeme pouze měnit parametry u pod-příkazů, např. rovně 3; doprava 90° apod.

Žáci si mnohdy nedokáží uvědomit, jakou část programu mohou opakovat. Někteří dokonce ani z vytvořeného programu bez tohoto příkazu nepoznají, která část se opakuje pořád dokola. Považuji tedy za nutné tomuto příkazu věnovat mnohem více času a dohledu nad žáky, aby program nevytvořili bez příkazu „opakuj“ a neposunuli se na další úlohu.

Tento příkaz je specifický tím, že se přímo do něj vkládají jiné příkazy. Není to jednoduchý krok, který by se dal jen tak přejít. Musíme dbát na to, aby žáci opravdu vnořili další příkazy do příkazu „opakuj“.

Abychom poznali, zda skutečně můžeme použít tento příkaz, je nutné si celý program pořádně prohlédnout či nejprve předem důkladně promyslet. Vytvořit si představu, náčrt nebo klidně i celý program, aby bylo splněné zadání. Z vytvořeného programu lze poznat opakovací postup. Zaměříme se na jednotlivé kroky a sledujeme jejich sled, pozorujeme, zda se některé celky opakují a jdou za sebou. Tyto celky pak vložíme do příkazu „opakuj“ a navolíme počet opakování.

Pro příklad takové úlohy, jsem vybrala úlohu č. 17, která je na obrázku číslo 5.



Obr. 5 – Úloha č. 17; zadání: *Spoj všechny ostrovy. Postav mosty, ale neznič při tom všechnu trávu.*

Žák v této úloze má spojit ostrovy, nesmí však zničit na ostrovech trávu. Využívá se zde i příkaz „kresli/nekresli“ a je zde použit právě příkaz „opakuj“. Z obrázku číslo 5 je zřejmé, že jsou do něho vnořeny všechny další příkazy. U samotného příkazu

„opakuj“ je pak změněn parametr na číslo 4, tudíž se má celý vymyšlený program opakovat 4krát. Odstraníme tak složité a zdlouhavé zadávání stejného programu pořád dokola a celý program zjednodušíme a zkrátíme.

4.4 Řešíme problémy

V tomto oddílu žáci rozvíjí kritické myšlení. Mají za úkol vyřešit danou úlohu pomocí jejich vlastního uvažování. Je jen na nich, jak problém vyřeší, avšak pořád dbáme na to, aby program co nejvíce zkrátili a měli jej nápaditý. Mají zde např. dokreslit zuby žralokovi, při této úloze v první fázi pozorování se děti mnohdy snažily tuto úlohu obejít a žralokovi dokreslily méně zubů než má žralok na pozadí. Je tedy opravdu důležité správnost provedení úloh kontrolovat a smyslně vést výuku. Dobrou kontrolou mohou být i spolužáci samotní, pouze si dva mezi sebou vymění místo a zkontrolují susedovo řešení.

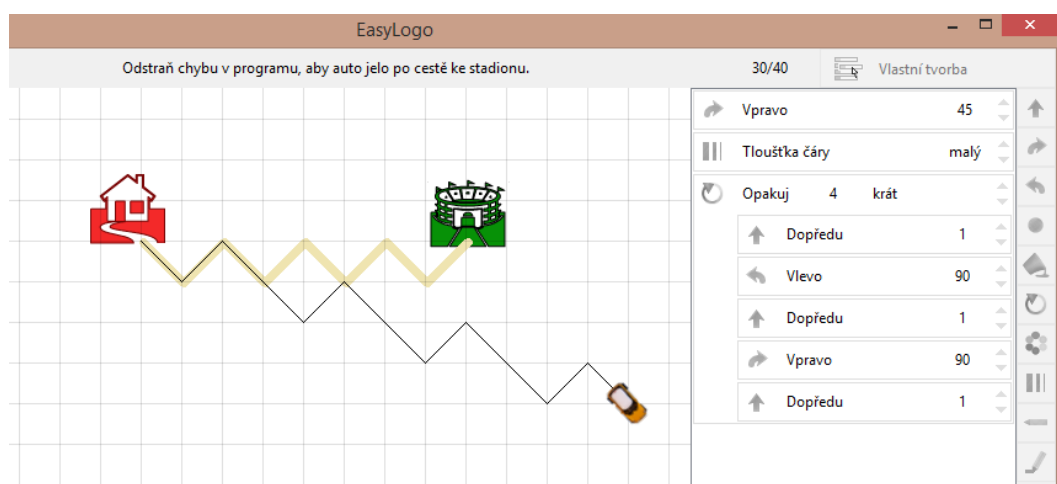
Mezi tyto úlohy se řadí například právě ta, kde mají žáci žralokovi dodělat zuby, nebo úloha, ve které mají za úkol dokreslit stromům kmeny.

4.5 Opravujeme chyby

Oddíl zaměřený na poznávání chyb v programu. Program už je předem vytvořený, ale pohyblivý tvar (př. želvy, auta) není v cíli cesty nebo není na svém místě. Úkolem je tak najít chybu v programu a změnit u daného příkazu parametry tak, aby byl problém odstraněn a úloha byla správně vyřešena, přičemž ale nesmí být z programu nic vyřazeno ani do něho nesmí být nic přidáno.

V tomto oddílu poznáme, zda žák opravdu programování rozumí. Hledá chyby ve vytvořeném programu, někteří žáci však mohou úlohu řešit pomocí metody pokus/omyl a bezmyšlenkovitě měnit parametry u použitých příkazů. Je tedy třeba jejich stálá kontrola.

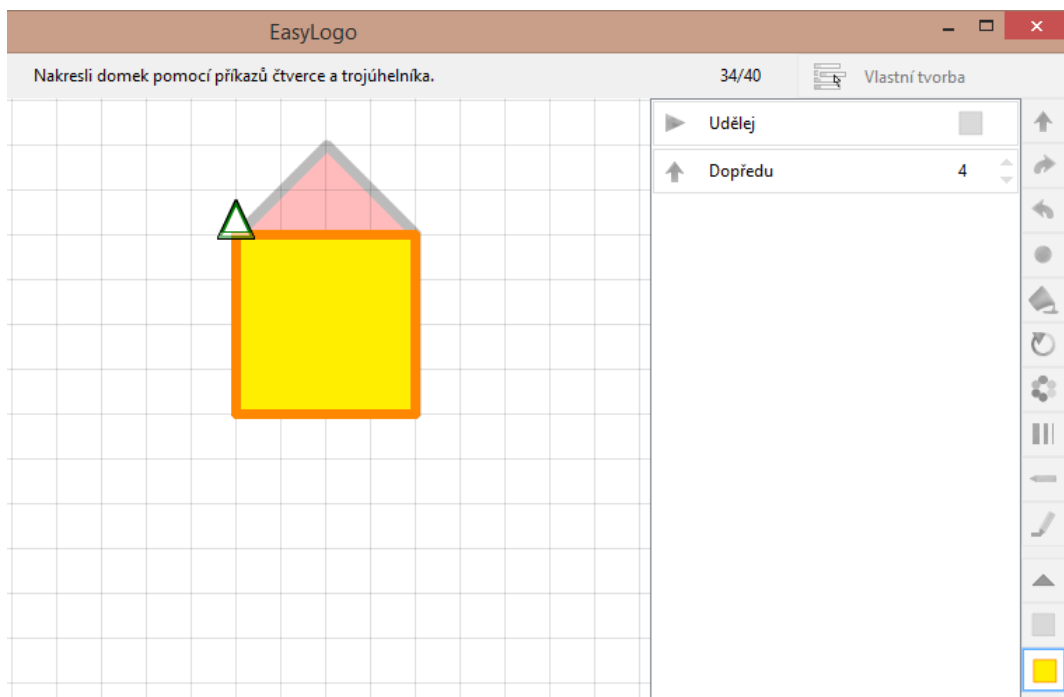
Na obrázku s číslem 6 vidíme jednu takovou úlohu, kde žáci mají za úkol najít a opravit chybu v programu. V této úloze mají žáci zjistit, že je zde poslední vnořený příkaz, tj. „dopředu“, zcela zbytečný a je tak v programu navíc. Mají ho tedy odstranit a program bude správně.



Obr. 6 – Úloha č. 30; zadání: *Odstraň chybu v programu, aby auto jelo po cestě ke stadionu.*

4.6 Procedury

Procedury z tohoto oddílu zahrnují práci s již hotovými tvary jako je např. čtverec či trojúhelník. Žáci mají za úkol z nich vytvořit dům, poté panelák, ulici a pak i celé náměstí. Zároveň mohou operovat i s hotovým tvarem, změnit parametry v proceduře vytvářející určitý tvar, např. právě čtverec. Na obrázku č. 7 vidíme rozpracovanou jednoduchou úlohu s procedurami, kde se právě pomocí čtverce a trojúhelníku vytváří domek.



Obr. 7 – Úloha č. 34; zadání: Nakresli domek pomocí příkazů čtverce a trojúhelníka. – rozpracovaná úloha.

5. Prvotní pozorování postupu výuky v EasyLogo na ZŠ

Na úplném začátku mé práce jsem se vydala do základních škol v Českých Budějovicích a jejich okolí, kde jsem buď jen pozorovala hodiny informatiky, nebo je sama odučila. Pozorovala jsem žáky při jejich práci, povídala si s nimi a poté následovaly rozhovory s učiteli o postupu výuky. Prováděla jsem analýzy jednotlivých úloh, ze kterých pak vyplývala potřeba dalších navazujících úloh, či poupravení stávajících.

5.1 Pozorování výuky programování na ZŠ v prostředí EasyLogo

V hodinách informatiky jsme se věnovali výuce programování právě prostřednictvím výukové aplikace EL. Zapisovala jsem si poznámky, připomínky a komentáře k probíhající výuce, k reakcím žáků, k pochopení úloh. Po skončení hodiny jsem celý průběh konzultovala ještě s přítomným učitelem a vyslechla jsem si i jeho dojmy z práce s touto aplikací.

V pozorování mě velice zaujal zájem žáků o tuto výuku. Bylo to pro ně něco jiného, nového a neotřelého, aplikaci neznali a ve velké míře si ho rychle oblíbili. EL je pro ně příjemné svým jednoduchým ovládním, v prvních úlohách pouze prostřednictvím směrových šipek a v dalších pak přetahování jednotlivých příkazů pod sebe a dále i tím, že okamžitě vidí výsledek svého jednání.

Závěrem pozorování bylo zpracování získaných informací z výuky informatiky. Sestavila jsem si plán postupu při vyučování jednotlivých příkazů a začala jsem pracovat na nové sadě úloh.

Původní sadu úloh jsem prověřovala v pátých ročnících základních škol. Hodiny jsem si převážně odučila sama, avšak v některých případech jsem pouze konzultovala výuku EL s učiteli. Při pozorování žáků jsem dávala pozor na správnost provedení úlohy, na porozumění zadání a na plynulost učení se novým příkazům.

Z tohoto pozorování mi vyplynulo, že žáci nerozuměli některým úlohám kvůli nepřesnostem v zadání, dále pak, že v některých úlohách by žáci mohli použít příkazu, který je chceme naučit, velmi snadno obejít, tudíž zde bylo potřeba upravit pozadí a pak také vyplynulo to, že mezi některými úlohami jsou až moc velké kroky co se týče pochopení vyučovaného příkazu. Žáci by zde potřebovali více času na procvičení.

5.2 Reakce žáků

V první fázi pozorování jsem se snažila vytipovat si žáky zdatné v informatice, průměrné žáky a žáky slabší. Ráda bych se zmínila zejména o jedné málotřídní škole, kde na hodině informatiky bylo pouze 8 dětí, a proto jsem mohla vyslechnout názory všech dětí bez problému, protože na práci s tímto prostředím jsme měli spoustu času a kontrola práce zde byla velmi jednoduchá a rychlá. Vyskytoval se zde jeden velmi zdatný žák, co se informatiky týče, a naopak jeden opravdu slabý žák, tudíž bylo dobré v této skupině pozorovat jejich počínání při práci s PC, potažmo s aplikací EL.

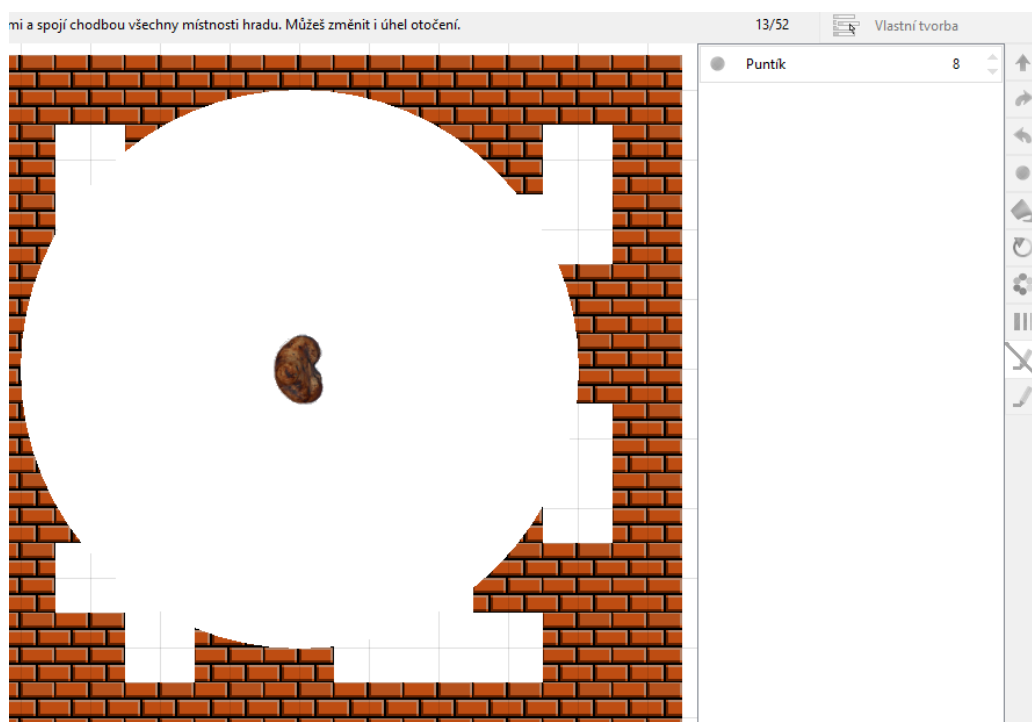
Onen zdatný žák má samozřejmě doma vlastní PC, což v dnešní době není ničím zvláštní, ba naopak začíná to být nepsaným pravidlem. Stane se, že se učitelé informatiky na prvním stupni základních škol, kdy předmět informatika není jejich hlavním předmětem, mohou setkat se situací, kdy žáci budou při této práci v některých programech či aplikacích zdatnější než oni samotní. Pak se samozřejmě ve třídě, jako právě v této, vyskytují i žáci, kteří nemají doma přístup k PC, a jedinou práci s ním odvádějí ve škole. Zde jsem měla jednu takovou žačku.

Při prvotním seznámení s aplikací se mým žákům rozzářili oči a podle jejich slov to bylo „fakt hustý“. Společně jsme si do PC stáhli toto výukové programovací prostředí a otevřeli jej. Žáci si vyslechli instrukce a společně jsme pokračovali dál. V této třídě je z osmi žáků šest chlapců a dvě dívky a hlavně tedy mezi chlapci panuje určitá soutěživost a rivalita, i proto se snažili mít úlohy co nejrychleji a správně hotové.

V prvním oddílu jsme postupovali vcelku rychle, vzhledem k věku žáků, jejich komentáře k úlohám byly: „to je lehký“, „fakt primitivní“, „to je easy“. Zde s námi ještě držela tempo i slabší žačka, u které jsem samozřejmě dbala na větší zřetel, co se týkalo kontroly.

Úspěch sklízela úloha s číslem 13, kde má Golem projít zdmi a spojit tak všechny místnosti jednou chodbou. Když zde žáci dostali pokyn, že má být úloha splněna na co nejméně příkazů, byli nadšení. Žáci se zde předháněli, kdo má kratší program a úloha nám tak zabrala více času. Nicméně jsem žáky nechala, aby mohli použít svou logiku a pokusit se o co nejlepší řešení. Neoriginálnější „řešením“ této úlohy bylo, kdy jeden žák (silnější ze třídy) objevil příkaz „puntík“ ještě než jsme se s ním stihli seznámit, a použil ho při této úloze. Na jeden příkaz tak spojil všechny místnosti a tím byla naše práce s touto úlohou ukončena. O řešení úlohy se však nejedná, jedná se jen

o jakýsi momentální nápad, protože se v tomto případě žák neučí programovat. Bylo tedy nutné žáky upozornit na to, že to není řešení úlohy. Žákův počín je vidět na obrázku č. 8.



Obr. 8 – Úloha č. 13 – originální „řešení“.

Celkově jsme se s touto třídou při prvotním testování ve výuce této aplikaci věnovali 6 vyučovacích hodin, avšak ne vždy to bylo celých 45 minut. Za tuto dobu jsme společně prošli všechny úlohy, mnoho z nich jsme dělali hromadně, kdy si žáci mohli prohlédnout řešení úloh na projekci a společně jsme si toto řešení vysvětlili a odůvodnili.

Takto probíhalo pozorování téměř na všech vybraných školách, jen s tím rozdílem, že ve třídě nebylo pouze osm žáků a byla tak usnadněná kontrola, nýbrž zde bylo až sedmadvacet žáků, což v některých třídách znamenali dva žáci na jeden PC.

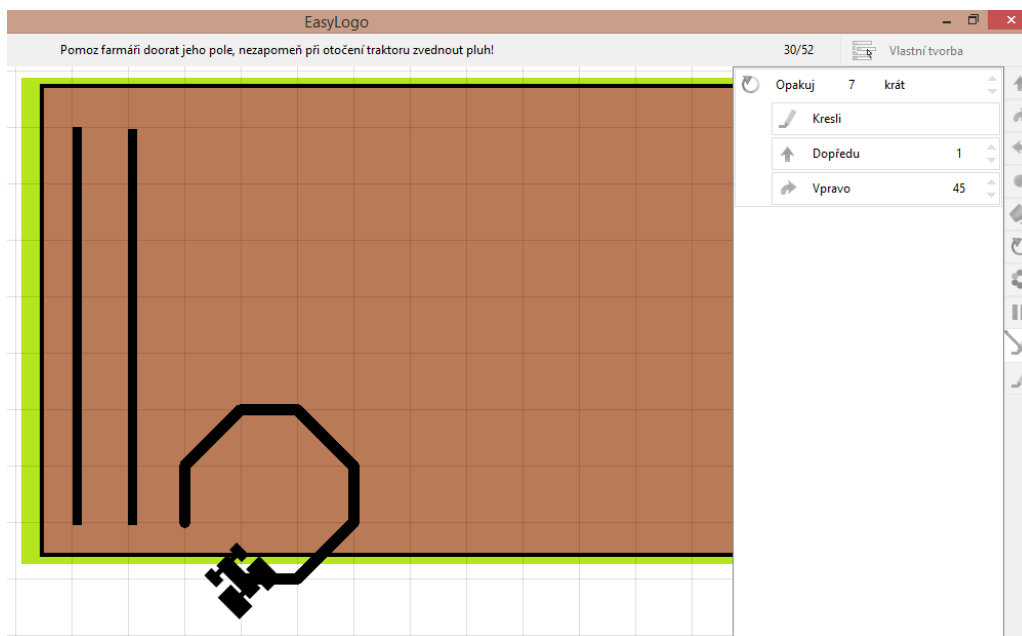
5.3 Žákovské chyby při řešení úloh

Ze svého pozorování jsem vybrala zajímavé žákovské chyby, které bych ráda představila v této podkapitole.

První chybu udělala v mé praxi jedna ze slabších žákyň. Chybu se nám podařilo vysvětlit pomocí dialogu ve třídě, kdy se do vysvětlování zapojili samotní žáci.

Chyba se vyskytla v úloze č. 30 (nová sada), kde mají žáci za úkol pomoci doorat pole farmáři, přičemž na konci řádky mají vždy zvednout pluh.

Tato úloha je zaměřena na příkaz „opakuj“ a je sama o sobě dost náročná. Žáci si musí uvědomit, které příkazy musí vnořit do příkazu „opakuj“ aby byl program vytvořen správně. Chyba žákyně je vidět na obrázku číslo 9.



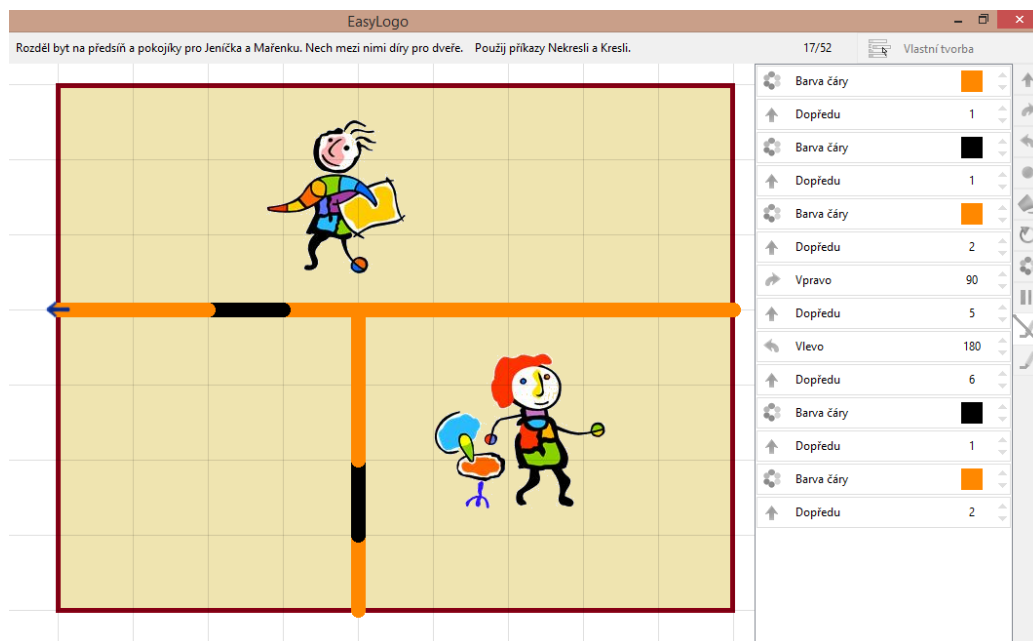
Obr. č. 9 – Úloha č. 30; zadání: Pomoz farmáři doorat jeho pole, nezapomeň při otočení traktoru zvednout pluh! – Žákovská chyba.

Dívka správně zařadila do programu příkaz „opakuj“ a vnořila do něj příkaz „kresli“, „dopředu“ a „vpravo“. Nicméně, po tom, co do programu vnořila příkaz „dopředu“, změnila parametry u příkazu „opakuj“, aby se dostala na konec řádky (pole). Ovšem parametry měla změnit u příkazu „dopředu“. Proto se při vnoření příkazu „vpravo“ zdeformovala dráha traktoru do různých tvarů, které se měnily při změně parametrů u otočení.

Žáci si úlohu mezi sebou vysvětlili, že když používají příkaz „opakuj“ musí nejprve vnořit všechny příkazy, které do něj spadají, a pak teprve mohou změnit parametry u samotného „opakuj“. Bylo opravdu skvělé je pozorovat, při takové diskusi.

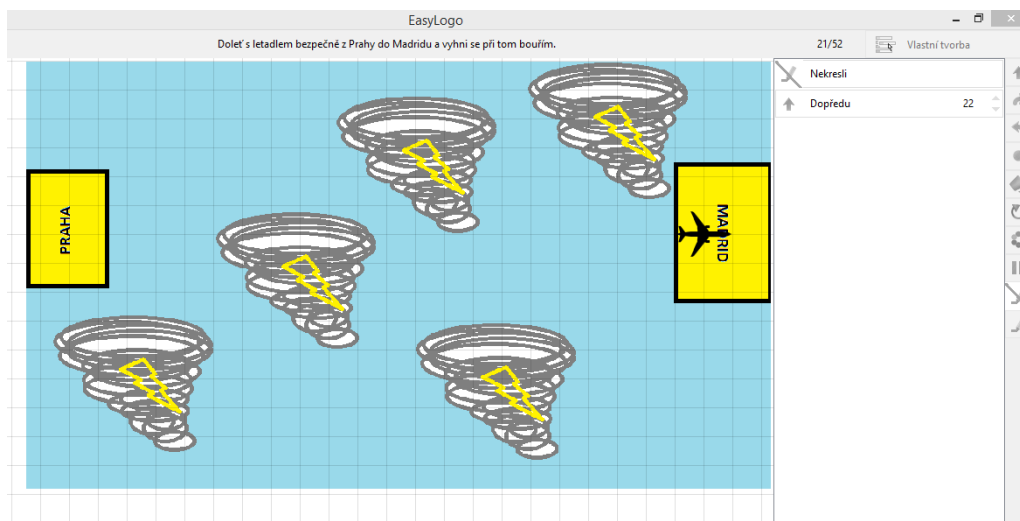
Další „chyba“, kterou chci představit, nastala v úloze č. 17, kdy mají žáci za úkol rozdělit byt na dva pokoje pro děti a předsíň, přičemž mají mezi stěnami vynechat díry na dveře. Chyba ani tak není chybou programátorskou, jako chyba v nepozornosti

žáka. Žák správně vytvářel program, avšak nevynechal mezi zdi díry pro dveře, nýbrž pouze změnil barvu čáry. Předcházející úlohy přibližují žákům příkazy „kresli/nekresli“, proto se předpokládá, že ho žáci použijí i v této úloze (dokládá to i zadání úlohy, kde mají žáci přímo napsáno – Použij příkazy „Nekresli a Kresli“). Nicméně jsem se v mé praxi setkala s tím, že žáci nevěnovali pozornost tomu, co dělali v předchozích úlohách a tento úkol se snažili vyřešit po svém, tak, jak můžeme vidět na obrázku č. 10.



Obr. 10 – Úloha č. 17; zadání: Rozděl byt na předsíň a pokojíky pro Jeníčka a Mařenku. Nech mezi nimi díry pro dveře. Použij příkaz Nekresli a Kresli. – žakovská chyba v řešení.

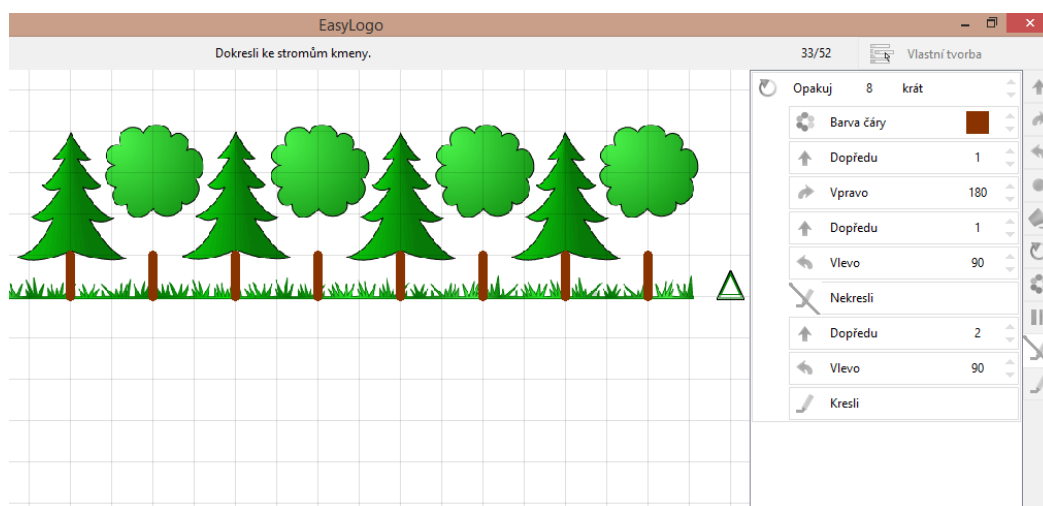
S další chybou jsem se setkala v úloze č. 21. Úkolem je doletět s letadlem z Prahy do Madridu a vyhnout se přitom bouřím, které jsou na pozadí. Žáci zde procvičují skládání programu a zároveň i změnu úhlu o 45°. Žakovou chybou bylo to, že zde, v této úloze, použil příkaz „nekresli“ a tím zneviditelnil dráhu letu, pak použil jen příkaz „dopředu“ a změnil parametry, aby se dostal do cíle. Jeho řešení úlohy můžeme vidět na obrázku č. 11.



Obr. 11 – Úloha č. 21; zadání: *Doleť s letadlem z Prahy do Madridu a vyhni se při tom bouřím.* – *Žákovská chyba v řešení.*

Chyba v řešení je to proto, že žák nesplnil zadání. Pokud bychom nepoužili příkaz „nekreslí“ a byla by vidět dráha letu, mohli bychom vidět, že letadlo se rozhodně bouřím nevyhnulo. Tato úloha může mít několik řešení, avšak toto mezi ně rozhodně nepatří.

I v úloze č. 33 se vyskytla u žáků opakovaná chyba. Zde mají za úkol dokreslit stromům kmeny. I zde se používá příkaz „opakuj“ a je tak nutné, aby si žáci uvědomili, co všechno musejí do programu vnořit.



Obr. č. 12 – Úloha č. 33; zadání: *Dokresli ke stromům kmeny.* – *Žákovská chyba.*

Na obrázku č. 12 vidíme častou chybu u žáků v této úloze. Někteří z nich si ji dokázali v programu opravit hned, jak viděli nesprávnost provedení, ale našli se

i tací, kteří potřebovali vysvětlení. Chyba spočívá v tom, že se kmeny u listnatých stromů nedotýkají koruny, jsou tedy stejně dlouhé, jako u jehličnatých stromů.

Výše popisované chyby jsou pouhým výběrem z řad dalších chyb. Je nutné si uvědomit, že se žáci pomocí této aplikace programování teprve učí, proto se musíme na možné chyby připravit a porozumět programu, abychom je dokázali, nejlépe společně se žáky, odhalit.

5.4 Závěr z prvotního pozorování výuky EL

Pozorování, zvláště pak na zmiňované málotřídní škole bylo zajímavou zkušeností a přineslo do mé práce kýžené ovoce, v podobě poznatků o potřebnosti dalších úloh či změně zadání. Při této práci mě pokaždé potěšil zájem žáků o EasyLogo, když měli radost ze soutěžení o to, kdo bude mít nápaditější či kratší program. Bylo na nich vidět, že to není pouhé bezduché poslouchání učitele, ale že je práce v prostředí EL doopravdy bavila a zároveň se mohli něco nového naučit.

Ujasnila jsem si, kde bych měla zvýšit počet úloh a které stávající úlohy poupravit. Ze začátku se jednalo o prvotní seznámení s aplikací, tedy o oddíl „Mód přímý“. Zde se mi zdálo, že by bylo potřeba vytvořit úlohy, které by žákům pomohly s prostorovou orientací v aplikaci a k uchopení jejích funkcí. Žáci totiž hrají na PC nejrůznější hry a ty mají pokaždé jiné ovládání. Ačkoli jim tedy mohou směrové šipky toto ovládání připomínat, v některých takových hrách se hráč po zmáčknutí směrové šipky vpravo okamžitě posune směrem doprava, kdežto ve výukové aplikaci EasyLogo se pohyblivý objekt pouze otočí v daném směru a pro pohyb se musí použít šipka dopředu.

Ze začátku každého oddílu práce zaostávala u pomalejšího děvčete. V prvotním seznámení s každým novým příkazem si byla nejistá a bála se neúspěchu. Pokaždé se nám však podařilo bariéru prolomit a i díky takto hravé formě každý oddíl respektive daný příkaz zvládla, protože ne vždy stihla dokončit všechny úlohy. Ovšem jednotlivé kroky si pamatovala a ve složitějších úlohách je dokázala použít.

V oddílu „Tvorba programu“ už se začaly více rozvíjet schopnosti žáků a jejich rychlost v plnění se výrazně lišila. Na začátku tohoto oddílu jsme si řekli, že program by měl být co nejjednodušší a neměl by obsahovat žádné zbytečné prvky, co by ho pouze zdelšily. Při první úloze, kdy za úkol bylo pomoci navléct korálky

holčičce, měla problém ze všech dětí pouze slabší žákyně. V další úloze, tedy v úloze s Golem, kdy má projít zdi a spojit tak všechny místnosti, se naplno projevila soutěživost chlapců ve třídě. Při práci s touto aplikací musí učitel, zvláště pak v některých úlohách takového typu, počítat s větším ruchem ve třídě a obrnit se na to trpělivostí. V tomto oddílu tedy bylo nutné přidat nové úlohy, aby byl zpomalen nástup na oddíl tvorby programu.

Dále se pak jednalo o úpravy vybraných úloh, při jejichž řešení často docházelo k nejasnostem. Žáci se setkávají v předmětu informatiky s novými pojmy často poprvé a tak se mnohokrát stalo, že v úloze, kde měli s včelkou (pohyblivý tvar) vyletět z okna a vrátit se zpět jen na jinou květinu, nerozuměli slovnímu spojení „vylet’ z okna“. Z toho důvodu zde byla jasná potřeba upravit zadání.

Zásadním a nejtěžším zlomem aplikace EL je oddíl úloh zaměřující se na příkaz „opakuji“. Zde měli, nejen na této škole, problém i zdatnější žáci a nejprve jsme si museli funkci toho příkazu vysvětlit, využívali jsme i křídlové tabule, kde jsme si vypisovali program a pak ho podrobněji zkoumali, jaké pasáže se v něm opakují a jaké příkazy do něj tedy můžeme vnořit.

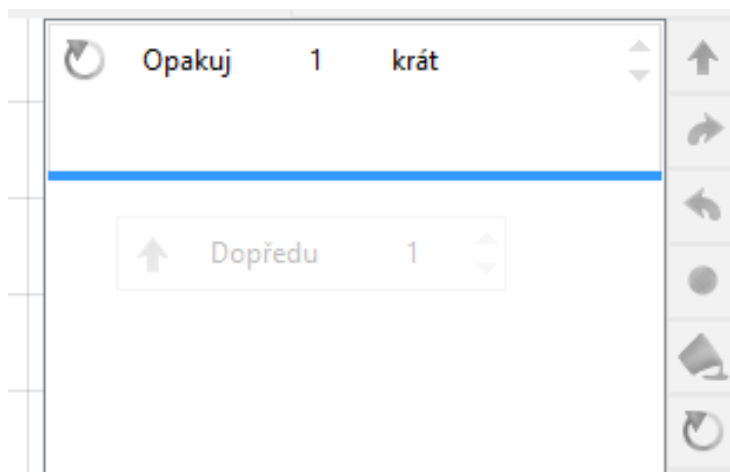
V případě příkazu „opakuji“ může být EL poněkud matoucí. Použijí zde příklad dalšího prostředí pro výuku programování a tím je aplikace Scratch. V tomto výukovém prostředí je hezky graficky znázorněno vnoření programu. Pokud používáme např. příkaz „opakuji“, který můžeme použít i v programu EL, jasně vidíme na obr. 13, že v aplikaci Scratch je příkaz hezky graficky připravený na vnoření dalšího příkazu, tedy v tomto případě např. posuň se o 10 kroků.



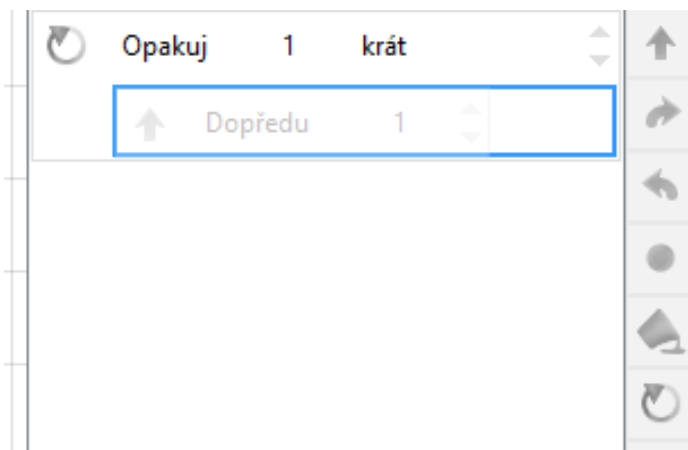
Obr. 13 – Prostor výukového programu Scratch, ukázka vnoření programu.

Zdroj: <https://scratch.mit.edu/>.

V EL takto jednoznačné „vnoření“ programu není. Pokud zde použijeme příkaz „opakuj“, musíme pak vnořený příkaz vložit přímo za něj, protože když posuneme kurzor myši byť jen o kousek níž, změní se poloha příkazu, který chceme vnořit a už do příkazu „opakuj“ patřit nebude (obr. 14). Je tedy třeba dávat si při práci s tímto příkazem větší pozor, aby do něj všechny náležící příkazy byly správně vloženy, jako je tomu na obrázku č. 15.



Obr. 14 – Špatné vložení příkazu do příkazu „opakuj“ – nyní není program vložený správně.



Obr. 15 – Správné vložení příkazu do příkazu „opakuj“ – nyní je program vložený správně.

Takto jsem se svými poznatky postupovala dál a dělala jsem si různé poznámky k jednotlivým úlohám, nákresy nových úloh a zapisovala jsem si všechny nápady pro změnu stávající výukové sady.

6. Tvorba nové sady úloh

Na základě pozorování a ověřování původní sady úloh jsem se pustila do vytváření úloh vlastních a do úpravy těch původních. Tento postup jsem měla již v návrhu pro postup vytváření výukového materiálu, který jsem si vytvořila pomocí metody ADDIE.

Jako první jsem si vytvořila seznam v příručním sešitě s přesným zadáním úloh a barevně u nich vyznačila použití různých příkazů. To mě navedlo na řazení některých úloh a na počet úloh v jednotlivých oddílech.

Dále jsem se věnovala nákresům nových úloh a jejich návaznosti na přidružené úlohy. A v neposlední fázi jsem tedy nově vytvořené, upravené i stávající úlohy seřadila za sebe, aby byl dodržen postup výuky jednotlivých příkazů. K výsledné fázi patří i seznam nově vytvořených aktivit, rozdělený do výukových oddílů.

V aplikaci jsem upravila i ikonu pro příkaz „nekresli“, kdy původní ikona byla ve formě zvednutého pera a nyní je ikona stejná jako u příkazu „kresli“ (pero připravené k psaní), nicméně je červeně přeškrtnuta. Dále jsem na úvodní stranu doplnila velkými písmeny nápis, aby si žáci před plněním úkolu pokaždé dobře přečetli zadání, protože v praxi jsem mnohokrát narazila na dotaz „A co máme dělat?“. Přitom stačilo jen zopakovat pokyn, aby si děti přečetly zadání a žáci hned věděli, podle jejich slov, co mají dělat.

Takto upravenou sadu jsem znovu vyzkoušela na základních školách, avšak ne ve stejných třídách, kde by testování postrádalo smysl, jelikož tam už žáci aplikaci znali, tudíž mohli předpokládat správnost řešení a využívat jiné příkazy.

6.1 Tvorba nových úloh pro aplikaci EasyLogo

EasyLogo je připravené pro tvorbu nových úloh. V souboru je přiložen návod na tvorbu nových úloh a teď je tvorba o to snazší, že byl vytvořen již zmíněný autorský program GEN. Vytvořené úlohy lze snadno uložit a zařadit mezi úlohy stávající a doplnit tak výukovou řadu o své nápady.

6.1.1 Ruční tvorba úloh

Tato podkapitola slouží i jako podrobný návod pro vytváření úloh pro výukovou aplikaci EasyLogo.

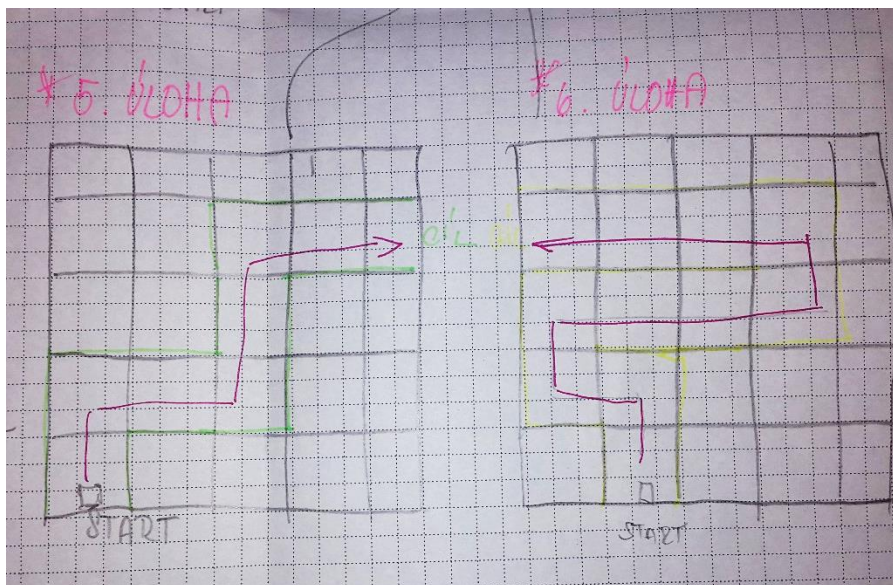
Pro mě byla tvorba nových úloh nelehkým úkolem. Zaprvé bylo potřeba nové úlohy vymyslet a promyslet smysl jejich užití v této výukové aplikaci a zadruhé jej správně naprogramovat, aby po spuštění EL vše fungovalo, jak mělo.

Tvořila jsem úlohy na začátek aplikace, kdy se s ní žáci teprve seznamují. Zde se používají pro ovládání daného tvaru pouze směrové šipky. Žáci si musejí uvědomit orientaci v programovacím prostředí, co se stane, když se tvar otočí, kdy se musí udělat krok vpřed atd. Proto jsem zde vytvořila dvě úlohy navíc, aby žáci tyto aktivity mohli snáze pochopit.

Dále jsem vytvářela úlohy, ve kterých je zapotřebí přetahovat příkazy do příkazového rámečku, první z mých takto vytvořených úloh je téměř shodná s jednou z úloh předchozích, avšak změna je právě tam, kde se musejí příkazy přetahovat. Úlohy mají podobný vzhled, aby žáci věděli, jak má úloha ve výsledku vypadat a tak je úloha ulehčena na pochopení. Následují úlohy původní až do úlohy č. 21, kde jsou za sebe přidány dvě nově vytvořené úlohy, které mají v neposlední řadě za úkol upevňovat přetahování příkazů do příkazového rámečku.

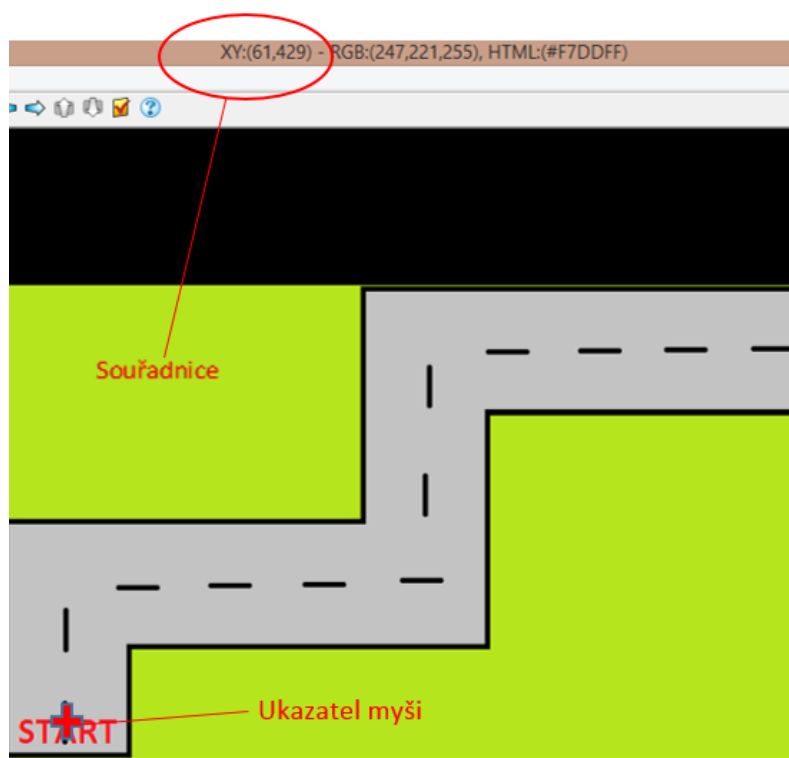
Takto vytváření úloh postupovalo dál, přidávala jsem zejména v místech, kde jsem si v původním pozorování ve výuce vytipovala možné nedostatky.

Před samotnou tvorbou úloh jsem je potřebovala promyslet, zda budou vhodné vzhledem k pohybu tvaru, který je omezen mřížkou. K tomu mi posloužil obyčejný čtvercovaný papír, kde vznikaly první návrhy nových úloh, dvě z nich jsou na obrázku č. 16.



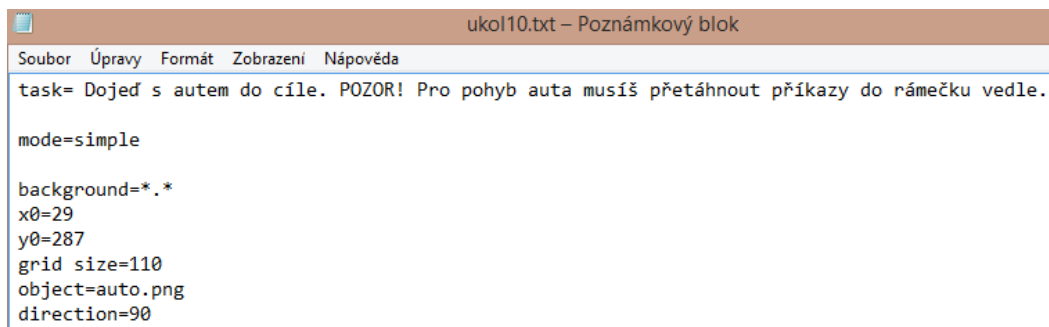
Obr. 16 – Náčrty nových úloh.

Samotné programování úloh pro mě, jako pro laika, bylo v některých chvílích velmi obtížné. Zejména v řešení usazení tvaru (př. auta) na pozadí, do správných souřadnic. Pro získání těchto informací jsem využila program IrfanView. Otevřela jsem si ze souboru vybrané pozadí a najela ukazatelem myši na požadované místo začátku tvaru. Zde jsem musela podržet stlačené levé tlačítko myši a v programu se mi ukázaly souřadnice tohoto místa, tak, jak je tomu na obrázku č. 17.



Obr. 17 – Souřadnice pohyblivého tvaru; program IrfanView

Tento a další údaje jsem musela přesně napsat do Poznámkového bloku a uložit je pod číslo úlohy. Pro ukázkou jsem vybrala program jedné z nově vytvořených úloh, konkrétně úlohu č. 10; zadání: Dojeď s autem do cíle. POZOR! Pro pohyb auta musíš přetáhnout příkazy do rámečku vedle. Její program obsahuje obrázek č. 18.



```
ukol10.txt - Poznámkový blok
Soubor Úpravy Formát Zobrazení Nápověda
task= Dojeď s autem do cíle. POZOR! Pro pohyb auta musíš přetáhnout příkazy do rámečku vedle.
mode=simple
background=*. *
x0=29
y0=287
grid size=110
object=auto.png
direction=90
```

Obr. 18 – Program úlohy č. 10 z nové sady

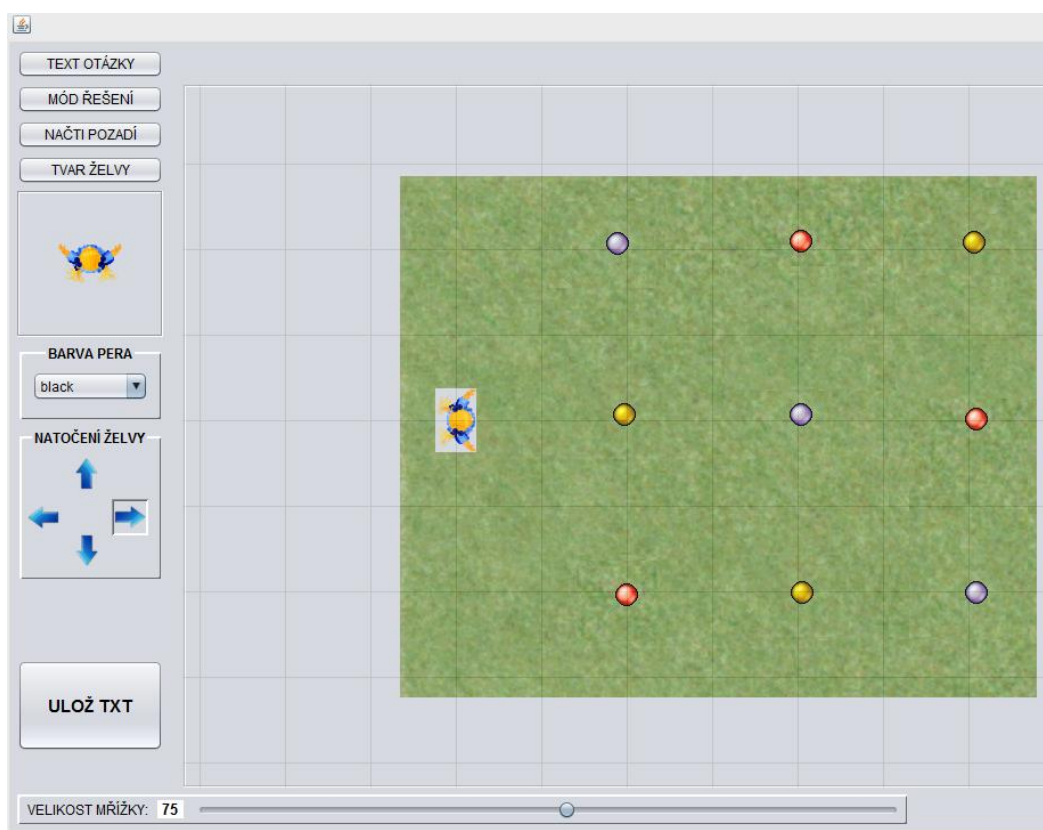
Ve vytvořeném programu vidíme, že zadání úlohy se píše jako první a musí být předepsáno slovem „task“ a následuje znaménko „rovná se“. Dalším údajem je mód (mode), ve kterém má být úloha řešena, zde je zvolen mód simple, neboli normální mód, při kterém je nutné přetahovat příkazy do příkazového rámečku. Následuje údaj „background“ což je pozadí úlohy. V této úloze je vyplněn údaj takto =*. *, což znamená, že pozadí je uloženo ve stejné složce, jako úloha a nese i stejný název, pouze se jedná nikoli o textový soubor, nýbrž o obrázek. Pod údaji „x0“ a „y0“ se skrývají ony souřadnice, které jsem zjistila pomocí programu IrfanView. Údaj „grid size“ značí velikost mřížky a následují údaje o objektu, což je objekt samotný (object), který musí být opět uložený ve stejné složce jako je úloha, a nakonec tedy jeho otočení (direction) v rámci mřížky.

6.1.2 Tvorba úloh pomocí autorského programu GEN

Při tvorbě úloh mi byl velkým pomocníkem již výše zmiňovaný autorský program GEN pro tvorbu úloh EL. Tento program jsem si stáhla do svého PC, kde jsem se však setkala s jediným problémem a to, že pokud v PC byl nainstalován program pro rozbalení souborů WinRAR a chyběl mi zde program Java, program nešel spustit. Nicméně po doinstalování programu Java byl GEN plně připraven k použití.

Prostředí tohoto programu je velice jednoduché na ovládání (obrázek č. 19). Na levé straně se nacházejí čtyři základní funkce:

- Text otázky – zde se píše zadání úlohy
- Mód řešení – na výběr je ze tří výše zmiňovaných módů (přímý, normální, procedurální)
- Načti pozadí – výběr pozadí pro úlohu
- Tvar želvy – výběr pohyblivého tvaru pro danou úlohu



Obr. 19 – Prostředí autorského programu GEN pro tvorbu úloh aplikace EasyLogo.

Z obrázku vidíme, že následující funkce jsou zvolení barvy pera, otočení želvy (tj. v jaké pozici bude pohyblivý tvar) a úplně dole se nachází lišta s posuvníkem, na které volíme velikost mřížky.

Po zadání těchto parametrů stačí jen kliknout myší na rámeček s nápisem „ULOŽ TXT“ a program vygeneruje textový soubor s hotovou úlohou. Tento text se musí zkopírovat a uložit do složky, kam se uloží i pozadí a pohyblivý tvar (pozadí i tvar musejí mít stejný název, který byl použit při tvoření úlohy, př. auto.png, jinak by se při spuštění úlohy v aplikaci EL nenačetly).

Program GEN se ke mně bohužel dostal až ve fázi, kdy jsem měla nové úlohy téměř hotové. Ale pro uživatele, kteří se rozhodnou sadu úloh dále rozvíjet, je program podle mého názoru velkým přínosem.

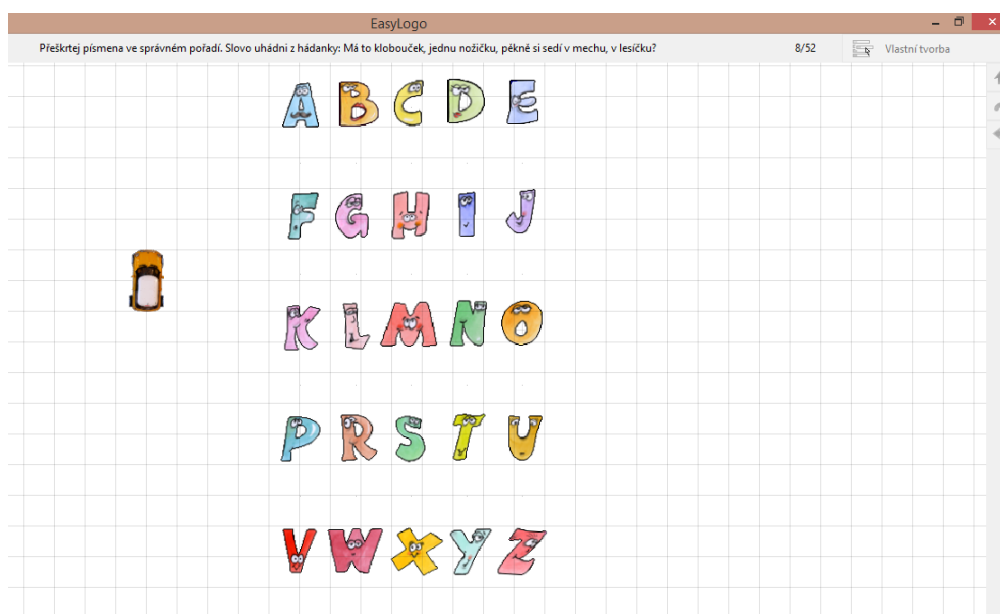
Pozadí úloh

Při tvorbě nových úloh jsem využívala již nahraná, původní pozadí nebo jsem si vytvářela sama svá pozadí v jednoduché a známé aplikaci Malování. Při vytváření těchto souborů jsem si musela zkopírovat velikost mřížky, kterou jsem chtěla použít a podle této velikosti vytvářet obrázek, aby se tvar pohyboval přesně v místech, které k tomu byly určené. Tato práce vyžadovala přesnost a tak se stalo, že jsem jedno pozadí dělala i několikrát a zabralo to více než hodinu času.

6.2 Upravené původní úlohy

Úloha č. 8

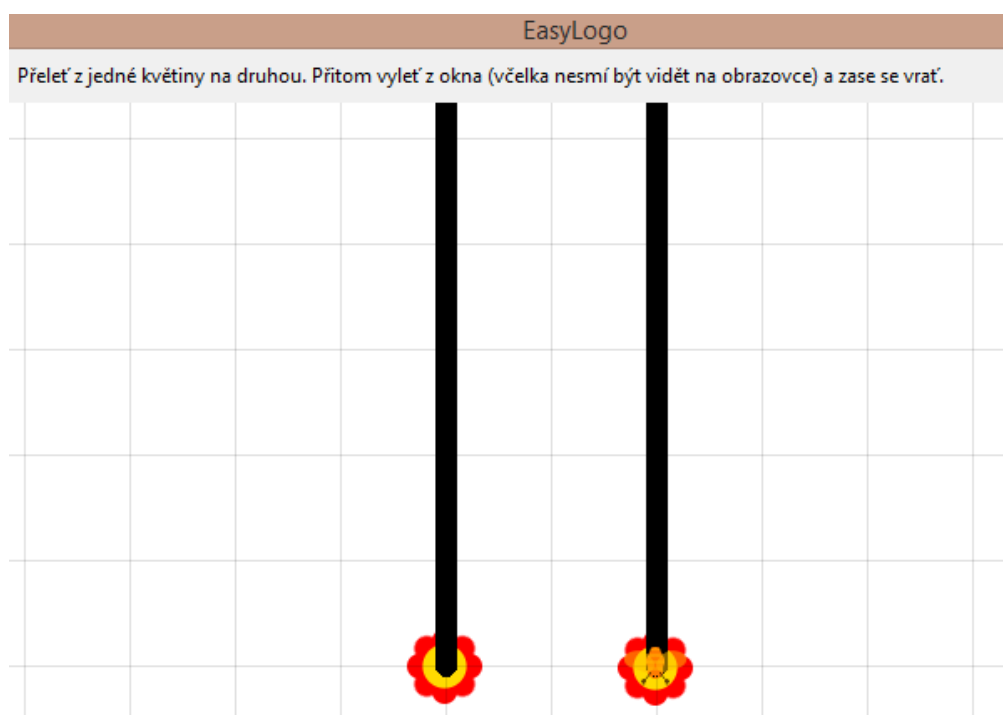
První poupravenou úlohou v sadě je úloha číslo 8 (obr. 20), kdy jsem převzala pozadí z úlohy předchozí a změnila její zadání. V předchozí úloze mají žáci za úkol přeškrtnat své jméno na tabulce písmen z abecedy, kdy jim ale zadání neuvádí, zda mají přeškrtnat písmena v pořadí, jak jdou ve jméně za sebou či nikoliv. V upravené úloze mají žáci za úkol uhádnout dané slovo z hádanky a jeho písmena pak přeškrtnat ve správném pořadí, kontrola tohoto úkolu je tak pro učitele velice jednoduchá. Úloha je zde zařazena hlavně pro rychlejší žáky, kteří zvládnou předchozí úlohy rychleji než ostatní. Učitel jim tak může zadat tuto úlohu, aby měli další práci navíc.



Obr. 20 – Úloha č. 8; zadání: Přeškrtej písmena ve správném pořadí. Slovo uhádni z hádanky: Má to klobouček, jednu nožičku, pěkně si sedí v mechu, v lesíčku?

Úloha č. 9

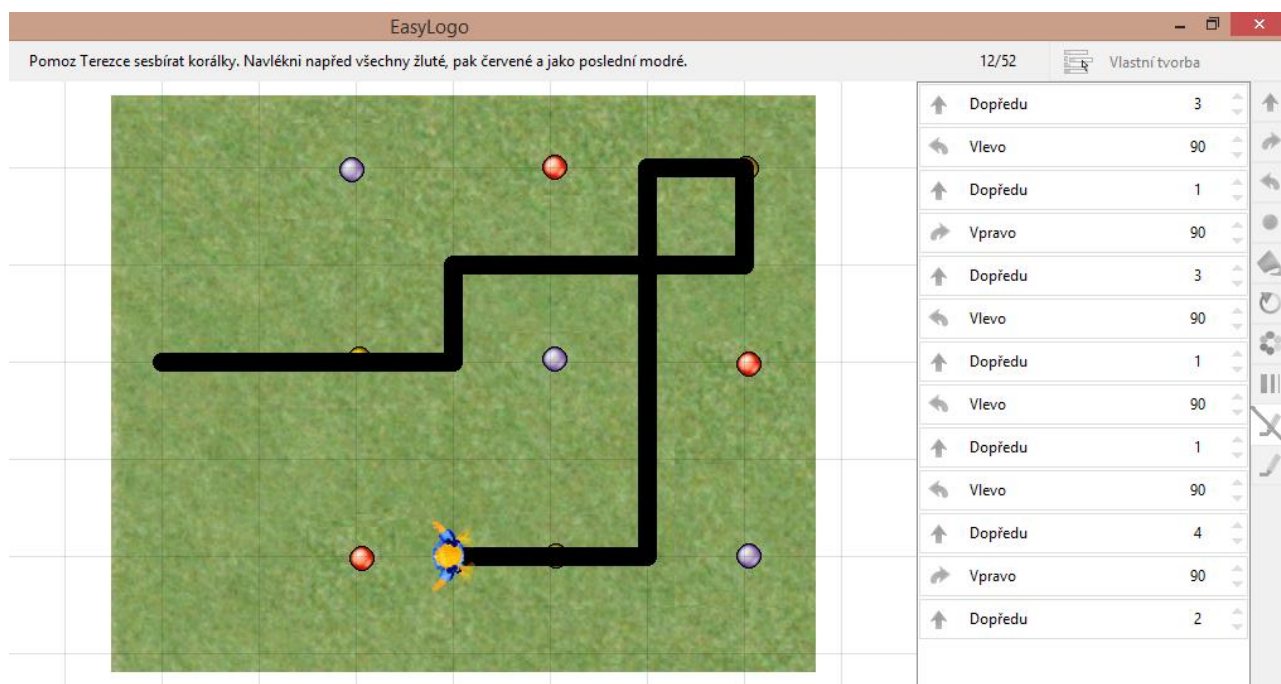
Devátá úloha v sadě je taktéž upravená, nicméně zde zůstává stejné pozadí i zadání, ovšem do zadání je přidána poznámka. Původní zadání této úlohy znělo: Přelet' z jedné květiny na druhou. Přitom vylet' z okna a zase se vrať. Úprava této úlohy byla dodána až po vyzkoušení nové sady, kdy mi nezávisle na sobě dvě paní učitelky řekly, že děti nechápou, co se myslí tím „vyletět z okna“. Do zadání byla tedy přidána poznámka (včelka nesmí být vidět na obrazovce). Bohužel toto upravené zadání ještě nebylo ve výuce vyzkoušeno či ověřeno, zda to pro pochopení žáků bude mít nějaký smysl. Úloha je vidět na obrázku číslo 21.



Obr. 21 – Úloha č. 9; Ukázka s novým zadáním – splnění úkolu „vylet' z okna“, který byl žákům nesrozumitelný.

Úloha č. 12

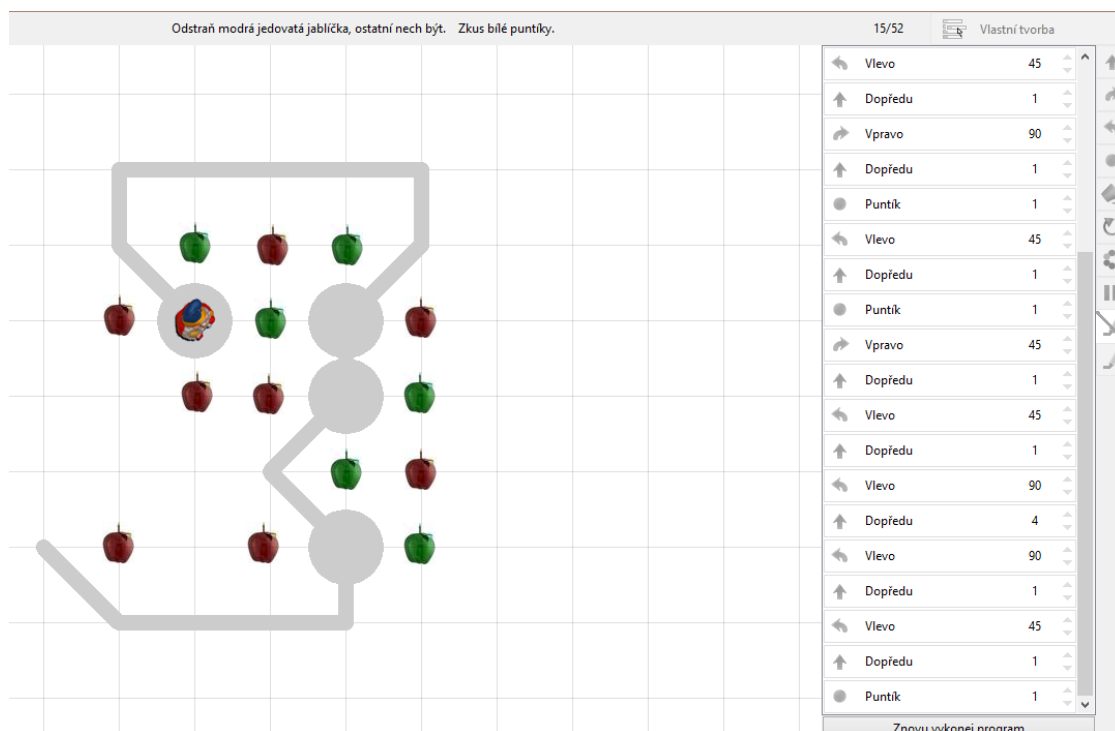
Zde je upravené pozadí z předešlé úlohy, kde má Terezka posbírat všechny korálky. Do upravené úlohy jsou přidány různobarevné korálky (červené, modré a žluté) a úkolem žáků je tyto korálky „posbírat“ ve správném pořadí. V této úloze je hlavním cílem procvičit přetahování příkazů do prostoru pro program a osvojení si této schopnosti. (obrázek 22)



Obr. 22 – Úloha č. 12 – rozpracovaný úkol, kde mají žáci za úkol posbírat korálky a dodržet přitom pořadí barev.

Úloha č. 16

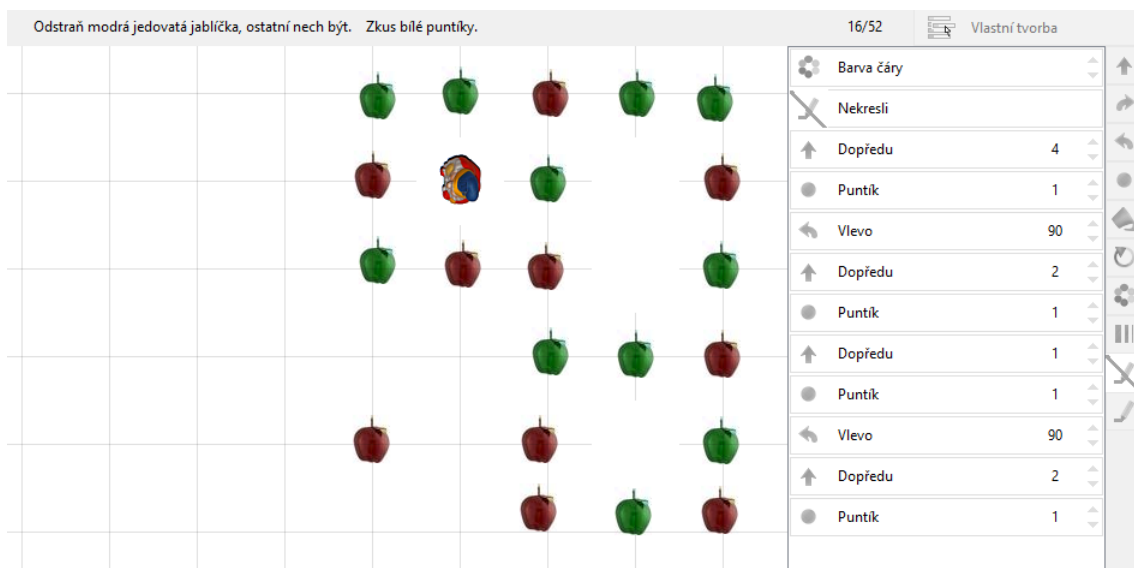
Úloha číslo 16 je opět úpravou své úlohy předchozí. V předchozí úloze lze zadání lehce obejít pomocí úhlopříček a nepoužít tak příkaz „kresli/nekresli“. Proto jsou do upravené úlohy v pozadí přidána další jablka.



Obr. 23 – Úloha č. 15; zadání: *Odstraň modrá jedovatá jablčka, ostatní nech být. Zkus bílé puntíky.* – *Nesprávné řešení.*

Na obrázku č. 23 můžeme vidět původní úlohu i s řešením (šedá barva je zde volena cíleně, aby byl vidět postup), řešení však obchází příkaz „kresli/nekresli“, který byl cílem této úlohy.

V upravené úloze na obrázku č. 24 je vidět, jak byla do pozadí přidána další jablka navíc, aby žák musel opravdu zmiňované příkazy použít. Nelze se dostat k modrým jablkům jinou cestou, než abychom museli použít příkaz „nekresli“ (nechceme, aby byla porušena jiná jablka než modrá), v prvním případě šlo obejít jablka pomocí úhlopříčky. Zadání úlohy zůstává stejné.



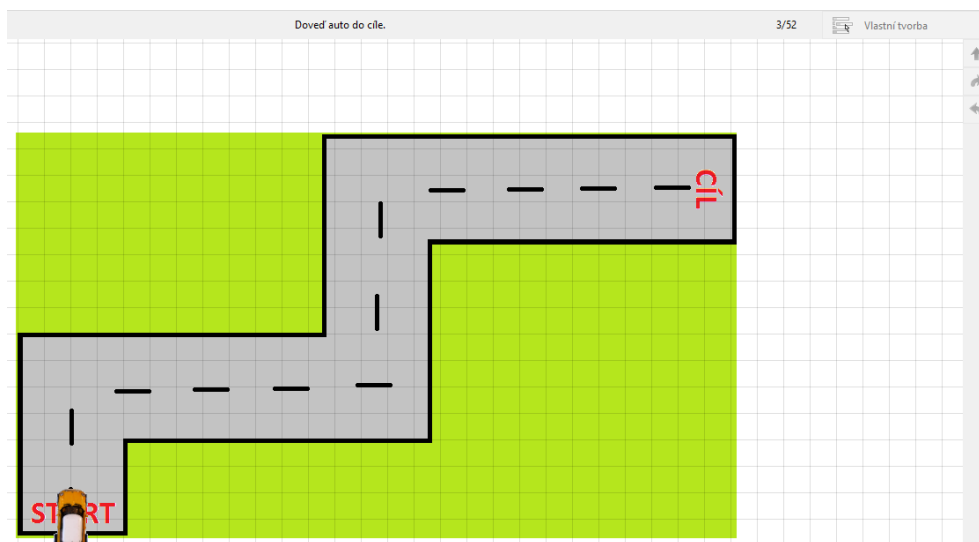
Obr. 24 – Úloha č. 16; Upravené pozadí, kam byla přidána jablka – řešení úlohy.

Tyto úlohy byly tedy pouze poupraveny v zadání či jejich pozadí, aby je žáci lépe pochopili, či si procvičili dané příkazy. V zadání se však po prvním testování nové sady vyskytly drobné nedostatky, které jsem se snažila při dalším odesílání napravit. Např. v úloze č. 7 zní zadání – „Přeškrtej písmena svého jména.“ Dostalo se mi zde zpětné vazby, že žáci nevěděli – „Jak přeškrtnat?“ Proto jsem se rozhodla pro další úpravu zadání, a jelikož je zde pohyblivý tvar v podobě auta, použila jsem zde následující znění – „Přejeď všechna písmena svého jména.“

6.3 Přehled nově vytvořených úloh

Úloha č. 3

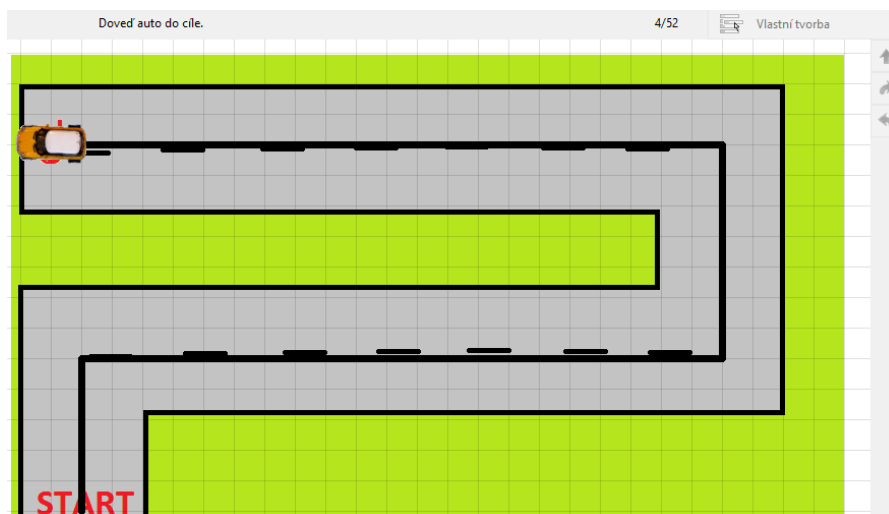
Pokud budu brát opět chronologické zařazení úloh nově vytvořených jako u těch upravených, pak první vytvořenou úlohou byla úloha s číslem 3. Zadání zní: „Doved' auto do cíle.“. Jelikož se jedná o jednu z prvních úloh, žáci zde ke splnění zadání využívají pouze směrových šipek. Tvar dráhy můžeme vidět na obrázku č. 25.



Obr. 25 – Úloha č. 3 – tvar dráhy.

Úloha č. 4

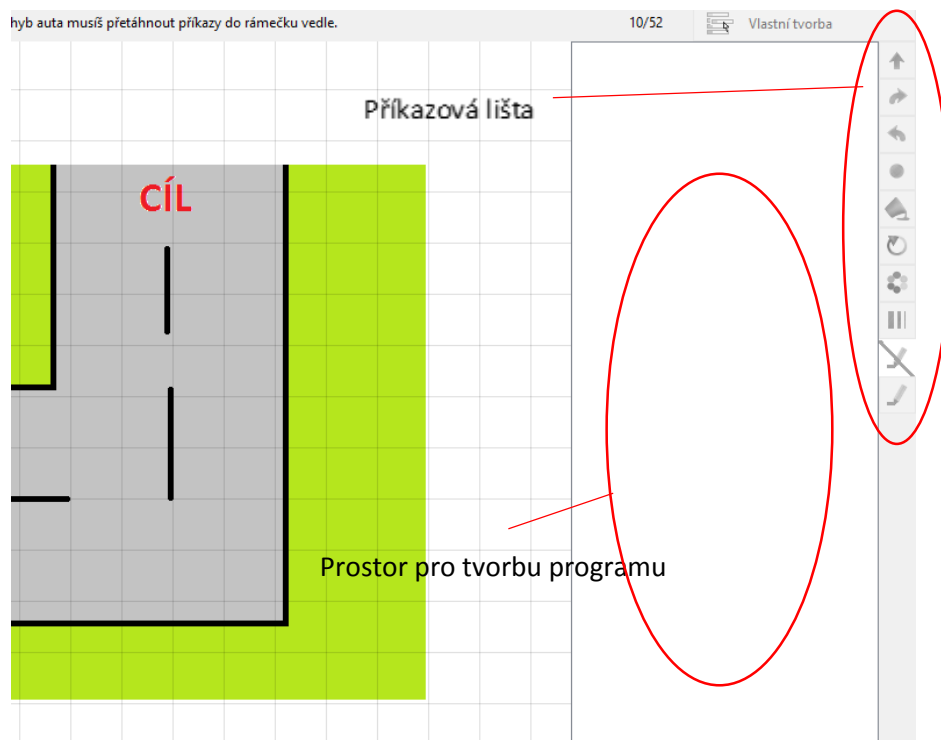
Následuje podobně vypadající nově vytvořená úloha (č. 4), kde je dráha do tvaru písmene „U“. Žáci zde stále procvičují a upevňují si pohyb tvaru auta pomocí směrových šipek. Zadání zůstává stejné jako u předchozí úlohy. (obrázek č. 26)



Obr. 26 – Úloha č. 4 – Dráha je do tvaru písmene „U“ – řešení úlohy.

Úloha č. 10

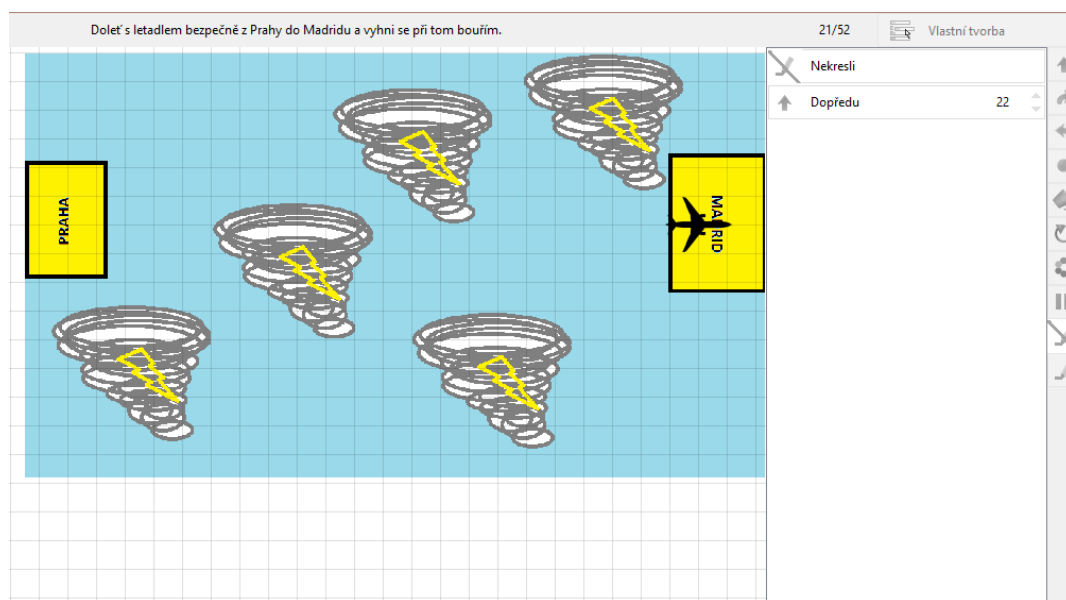
I třetí nová úloha (v sadě s číslem 10) je vytvořena ve stejném duchu jako dvě předchozí, nicméně tentokrát se neoperuje se směrovými šipkami, ale přechází se už na příkazovou lištu, tzn. na přetahování příkazů do prostoru pro tvorbu programu, jak můžeme vidět na obrázku č. 27.



Obr. 27 - Úloha č. 10 – ukázka příkazové lišty a prostoru pro tvorbu programu.

Úloha č. 21

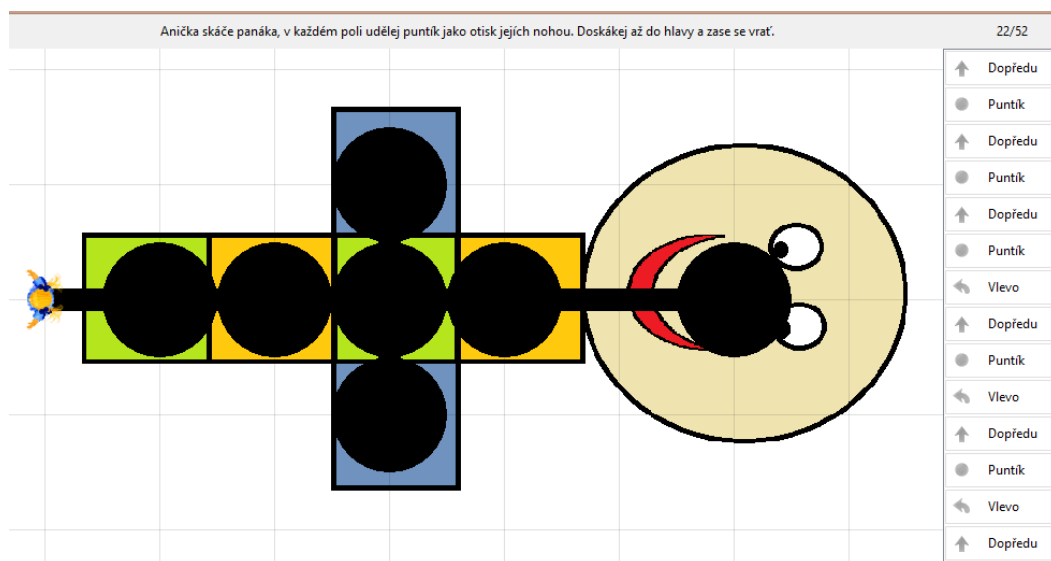
V úloze č. 21 se setkáváme s typem úlohy, kde procvičujeme přetahování příkazů do příkazového sloupce a zároveň si zde žáci mohou vyzkoušet otočení o jiný úhel než je 90° . Jedná se o úlohu, kde jsou na pozadí vyobrazena dvě letiště a mezi nimi je několik bouří (na obrázku jsou tornáda). Úkol zní, aby doletěli s letadlem z Prahy do Madridu a vyhnuli se přitom bouřím. V této úloze jsem se setkala i s originálním řešením, kdy žák pátého ročníku jako první použil příkaz „nekreslí“ a pak vložil příkaz „dopředu“, u kterého navolil příslušný počet políček, aby se dostal na protější letiště. (viz obr. 28) Považuji tedy za nezbytné tuto úlohu ještě nějakým způsobem upravit a to nejlépe ve specifikaci zadání.



Obr. 28 - Úloha č. 21 – Způsob řešení žaka pátého ročníku.

Úloha č. 22

Stejně jako úloha č. 21, tak i následující nově vytvořená úloha č. 22 procvičuje práci s příkazovým sloupcem, avšak navíc zde používáme puntíky. V pozadí je zde vytvořený obrázek panáka z dětských her, kdy ho děti musejí vyskákat a jako pohyblivý tvar je zde postava holčičky Aničky (obr. 29). Úkolem je doskákat až do panákovy hlavy a pak se vrátit na začátek. Přitom v každém poli zanechat puntík jako otisk holčinych nohou.

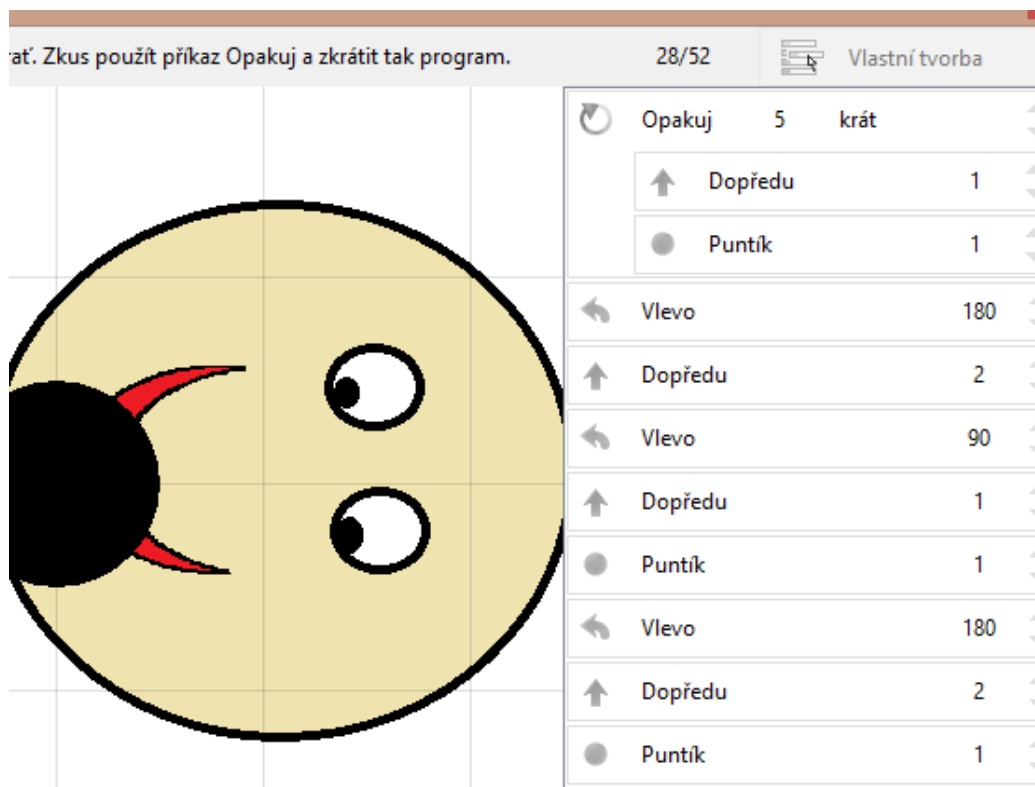


Obr. 29 – Úloha č. 22 – Ukázka pozadí s řešením úlohy.

Tyto dvě výše zmíněné úlohy (tj. č. 21 a 22), respektive jejich pozadí, jsou v sadě použity později ještě podruhé, ovšem s jiným typem zadání a procvičují jiný typ příkazů.

Úloha č. 28

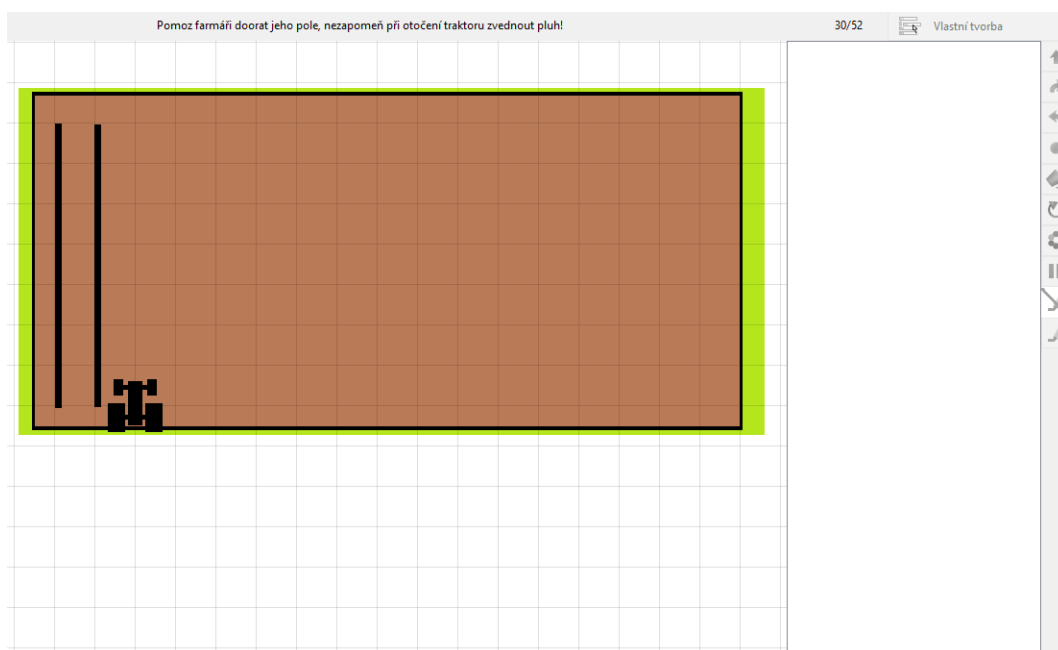
Již další v další nově vytvořené úloze je použito pozadí z úlohy č. 22, tj. obrázek panáka a pohyblivý tvar děvčete. I v této úloze je za úkol doskákat do hlavy panáka a zase zpět, ovšem tentokrát mají žáci za úkol použít příkaz „opakuj“. Jedná se totiž o úlohu č. 28, která se již řadí do oddílu věnujícího se tomuto příkazu. (Obrázek č. 30)



Obr. 30 – Úloha č. 28 – Použití příkazu opakuj.

Úloha č. 30

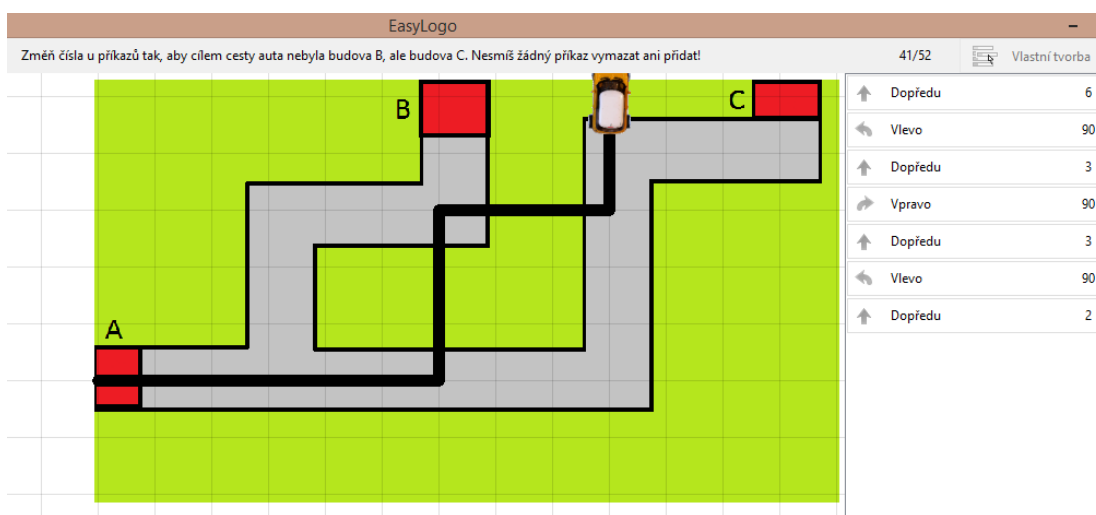
Na obrázku č. 31 je zobrazená nově vytvořená úloha č. 30, která je zaměřena na příkaz „opakuj“. Tato úloha je poněkud složitější, je nutné zde vytvořit delší řadu příkazů, které se vnoří do příkazu „opakuj“. Tuto úlohu při testování ve výuce zvládli jen bystřejší a výrazně rychlejší žáci, pro ostatní byla podle jejich slov „moc těžká“. I proto bych tuto úlohu ve výuce nepovažovala za nutnou k probrání, ale jako úlohu pro žáky, kteří rychleji zvládnou předchozí úlohy. Zde mohou předvést své logické uvažování a schopnost pracovat s aplikací.



Obr. 31 – Úloha č. 30; zadání: Pomoz farmáři doorat jeho pole, nezapomeň při otočení traktoru zvednout pluh!

Úloha č. 41

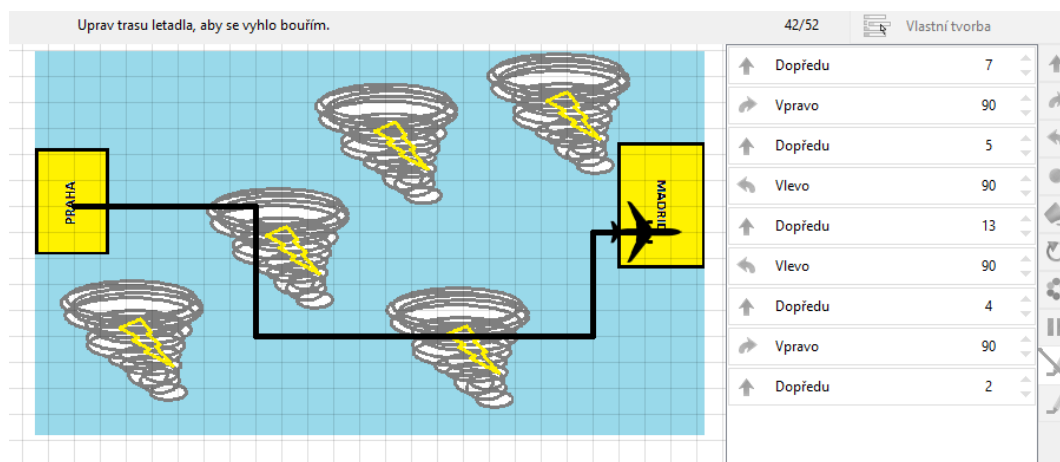
Předposlední vytvořená úloha je úloha č. 41 z celkových dvaapadesáti úloh. Tato je zařazena do oddílu „Opravujeme chyby“. Na pozadí je vyobrazen plán cesty a tři budovy – A,B,C. Původní trasa pohyblivého tvaru, tentokrát auta, byla z budovy A do budovy B. Zadání ale říká, aby žák upravil trasu tak, že cílem cesty auta bude budova s písmenem C. Při testování této úlohy jsme se žáky narazili na detail, že v zadání není napsáno, aby se z programu nic nemazalo a aby se do něj ani nic nepřidávalo. Tudíž někteří žáci vymazali celý původní program a vytvořili nový, přičemž stačí pouze změnit parametry u některých příkazů a úkol bude splněn. Právě proto jsem se rozhodla pro dodatečnou úpravu této úlohy a změnila její zadání. (Obr. 32)



Obr. 32 – Úloha č. 41 – Rozpracovaná úloha, kde je vidět pohyb při změně parametrů v původním programu.

Úloha č. 21

Poslední vytvořená úloha má v nové sadě číslo 42. Jedná se o stejné pozadí jako v úloze č. 21, jen je zde opět předem vytvořen program, jako v předchozí úloze s budovami tzn., úloha se také řadí do oddílu „Opravujeme chyby“. Letadlo má zde naplánovanou trasu skrz bouře a správné řešení je, změnit čísla u příkazů tak, aby se letadlo bouřím opět vyhnulo a dorazilo tak bezpečně do cíle své cesty. Úloha je vidět na obrázku č. 33



Obr. 33 – Úloha č. 42 – Původní program.

7. Ověření nové sady ve výuce

Po dotvoření nové sady EL jsem jej rozeslala učitelům informatiky na prvním stupni ZŠ. Cílovou skupinou byli převážně učitelé, kteří nemají s výukou programování moc velké zkušenosti. Další skupinou oslovených byly mé kolegyně z univerzity, které již učí na základní škole a které tento program znají v původním stavu. Samozřejmě jsem si i já sama vyzkoušela novou sadu EL v hodinách informatiky. Po tomto závěrečném testování jsem dostávala zpětné vazby od učitelů, kdy se osloveným učitelkám, bez zkušeností s EL, aplikace, jako náplň do informatiky líbila, avšak některé úlohy se zdály být pro žáky nejasné.

Oslovených učitelek jsem se posléze zeptala i na nově vytvořené či upravené úlohy. V mnoha případech se jim zařazení do původní sady líbilo. Jediná úloha, těžší než ostatní, byla úloha č. 30, kdy mají žáci za úkol doorat pole a použít příkaz „opakuj“. Tuto úlohu zvládli silnější žáci bez pomoci, s ostatními žáky se tato úloha musela udělat společně.

Další nedostatky, již zmiňované ve výpisu úloh, jsem po ozkoušení ve výuce odstranila. Tím byly ve většině případů chybné formulace zadání. Dále jsem i na apel učitelů přidala na úvodní stranu velký nápis, aby si žáci před svou prací vždy pečlivě přečetli zadání. Tento krok však podle mého názoru bude zbytečný a žáci budou dále používat dotazy typu „A co máme dělat?“.

Poslední úpravou v EasyLogu po vyzkoušení ve výuce byla změna ikony pro příkaz „kresli/nekresli“. Na tento problém mě upozornil vedoucí mé práce a po rozhovoru s učiteli, jsem zjistila, že by taková změna byla vítaná. Původní ikoně žáci nerozuměli a v řadě případů hledali příkaz „nekresli“.

Ohlasy na sadu EL byly tedy většinou kladné a EL pro oslovené pedagogy bylo přínosem do jejich hodin informatiky. Tito učitelé neměli v tematických plánech zařazené programování, proto jsem jim EL doporučila do dalších let a téměř všichni mi „slíbili“, že programování v EL zakomponují do své budoucí praxe a k tomu jim dopomůže, alespoň prozatím, nově vytvořená sada úloh EasyLoga.

7.1 Ověření ve výuce na základních školách

Sadu úloh jsem poskytla do třech škol v Jihočeském kraji (ZŠ Františka Křižíka – Bechyně, ZŠ Besednice, ZŠ Nedabyle) a celkem ji vyzkoušelo 51 žáků pátých ročníků s pěti učiteli. Testování probíhalo v různém rozsahu hodin, avšak ne méně než 4 vyučovací hodiny. Za tuto dobu měli jak žáci, tak i učitelé možnost EasyLogo blíže poznat, prozkoumat ho a vyzkoušet nové úlohy. Učitelům jsem poskytla i nově vytvořený seznam úloh, který je rozdělen na příslušné oddíly, a vyznačila jsem jim v nich nově vytvořené úlohy. Seznam úloh je přílohou této práce.

Z ověření nové sady vyplynuly poznatky týkající se nových úloh. Například, že v nových úlohách nebylo dostatečně popsáno zadání nebo byl úkol na žáky moc složitý. Problém se složitostí se týkal třeba úlohy č. 30, kde mají žáci za úkol pomoci farmáři s dooráním pole. Jedná se o úlohu, která slouží k procvičení příkazu „opakuj“. Na základě těchto připomínek ze strany žáků ale i učitelů jsem úlohu zařadila na konec oddílu, aby případně sloužila jako úloha pro žáky, kteří budou v práci rychlejší než ostatní.

Problém s nejasností u zadání se vyskytoval i v původních úlohách a tyto nedostatky jsem po tomto ověření odstranila. Zadání u úloh jsem ujasnila a přepsala.

7.2 Ověření učiteli

Svou práci jsem poskytla i do výuky na univerzitě, kdy se do role žáků vžili studenti programu Učitelství pro 1. stupeň ZŠ (20 studentů). Tito studenti mi poskytli zpětnou vazbu k nově vytvořeným úlohám. Jednou z připomínek byla k úloze číslo 12 (zadání: „Pomoz Terezce sesbírat korálky. Navlékni napřed všechny žluté, pak červené a jako poslední modré.“) Tato úloha by šla podle jejich názoru ještě upravit. Jedním ze způsobů by byl rozdělit ji na více úloh, kdy by v každé z nově vzešlých úloh Terežka (pohyblivý tvar) sbírala korálky pouze jedné barvy. Tím by tedy vznikly další tři nové úlohy.

A následujícím nápadem bylo zadání úlohy doplnit o informaci, že mají korálky jedné barvy navléknout pomocí provázku té samé barvy, tzn. při sbírání žlutých korálků, by žáci museli změnit barvu čáry na žlutou, při červených na červenou a při modrých na modrou.

Za zpětnou vazbu jak od žáků, tak od učitelů, potažmo studentů učitelství, jsem nesmírně vděčná. Tuto výukovou aplikaci budu ráda využívat ve své budoucí praxi, tudíž si mohu dále úlohy zdokonalovat a vylepšovat, popřípadě poskytovat dalším z řad učitelů do výuky informatiky.

7.3 Metodika výuky

Na základě pozorování ve výuce, jak prvotním, tak i závěrečném, jsem sestavila vlastní metodiku výuky k výukové aplikaci EasyLogo.

EasyLogo určen pro výuku programování se člení na sedm oddílů, zaměřujících se na jednotlivé příkazy, a v tomto sledu by se výuka měla dodržet. Tyto oddíly jsou začleněny do osnovy výuky.

7.3.1 Osnova výuky

1. Seznámení s prostředím výukové aplikace EasyLogo
2. Mód přímý (pohyb v prostředí s pomocí směrových šipek)
3. Tvorba programu (vytváření programu přetahováním příkazů do rámečku)
4. Čárkovaná čára (využití příkazů „kreslí/nekreslí“)
5. Opakuj (seznámení s principem opakování)
6. Řešíme problémy (řešení situací na základě naučených dovedností pomocí vlastní intuice)
7. Opravujeme chyby (hledání chyb v programu)
8. Procedury (využití již hotových tvarů)

Odůvodnění zvoleného postupu výuky

Seznámení s prostředím

Před začátkem práce s tímto prostředím je nutné, aby se s ním žáci dobře seznámili. Vysvětlíme jim proto co je to programování, proč se používá. Dále pak, že cíl programování je, vymyslet a sestavit co nejjednodušší a nejkratší program.

Při otevření EasyLoga by si žáci měli prohlédnout úvodní stranu, vyzkoušet všechna popsaná tlačítka a přečíst si jejich popis.

Mód přímý

Jako první oddíl byl zařazen právě tento, žáci se zde učí pracovat v prostředí této aplikace, uvědomují si princip mřížky (jeden čtvereček je jeden krok) a otočení pohyblivého tvaru. Oddíl obsahuje více úloh, avšak ne všechny musí učitel ve výuce použít. Je třeba zaměřit se na nejslabšího žáka a jemu podřídít tempo. Můžeme tedy složitější úlohy z daného oddílu využít jako práci pro rychlejší žáky.

Tvorba programu

Dalším krokem ve výuce je vytváření programu, tedy jde o posun k samotnému programování. V tomto oddílu se žáci hlavně učí přetahovat příkazy z lišty do rámečku a skládat je za sebe tak, aby byl program správný.

Čárkovaná čára

Jedná se o oddíl, kde se využívá příkazů „kresli/nekresli“. Pro žáky jsou tyto příkazy těžké na pochopení, přirovnáváme je tedy k tomu, jako bychom kreslili po papíře, když nechceme kreslit, musíme zvednout tužku od papíru, když za sebou chceme nechat stopu, musíme ji zas na podložku položit.

Opakuj

Oddíl obsahující kouzelný příkaz „opakuj“, který nám pomůže zkrátit program. Žákům musíme v první řadě přiblížit princip opakování. Jako další krok musíme dbát na správné vkládání příkazů do příkazu „opakuj“, jak už bylo zmíněno.

Řešíme problémy

Testování naučených dovedností je při výuce důležité. Tato část by se dala považovat za jakési jejich prověření. Žáci zde mají pět úloh, které mají na základě naučených příkazů vyřešit.

Opravujeme chyby

V následujících úlohách mají žáci za úkol hledat v již hotovém programu chyby. Ukazuje se tak, zda žáci programování opravdu rozumí, pokud nepracují metodou pokus omyl. Vidí před sebou hotový program, ale skrývá se v něm chyba. Když půjdou po jednotlivých krocích, snadno chybu odhalí.

Procedury

Poslední oddíl obsahuje úlohy, v nichž se používají ke splnění úkolu již hotové tvary, např. čtverec a trojúhelník. Z těchto tvarů žáci staví dům, panelák, náměstí, atd. Stále se ale musí pohybovat po čtvercové síti.

Zařazení do vyučování

Pro práci s EL si nemůžeme vyhranit jen několik hodin. Vhodné je ho zařazovat častěji a třeba jen na část hodiny, kdy tématem hodiny může být i něco jiného. Žáci se u této aplikace uvolní, odreagují, ale zároveň se i něco naučí. Můžeme ji tak zařazovat v průběhu hodiny, kdy vidíme, že jsou žáci unavení, nesoustředění. Stačí jen říct EasyLogo úloha číslo „x“ a žáci už mohou jen pracovat. Můžeme je namotivovat různými odměnami, jako jsou známky za práci v hodině, plusy či se mi v praxi osvědčilo, že kdo byl hotov, mohl navštívit internetové stránky po dobu, než dokončí práci ostatní spolužáci.

Každý pedagog by měl při práci znát produkt, který ve své praxi používá a jaké jsou jeho výhody a nevýhody a ani EasyLogo v tomto ohledu není výjimkou, i přes jeho jednoduchost. Je nutné znát jednotlivé příkazy a dodržovat postup jejich jakéhosi odhalování pro žáky, aby úlohy neřešili jiným způsobem. Výhodou tohoto programovacího prostředí je, že žáci okamžitě vidí své počínání, tvar, např. auto, se jim po ploše okamžitě po použití příkazu pohne, což ovšem může být pro některé žáky i matoucí, a dále může být výhodou také to, že učitel snadno zkontroluje správnost provedení úkolu a splnění či nesplnění zadání. U některých úloh může být několik řešení a je dobré je se žáky prodiskutovat, přijít na to, které řešení je přijatelnější a proč. Měli bychom žákům v tomto ohledu předávat zpětnou vazbu a ne jen jejich řešení přejít s mávnutím ruky. Stále dokola jim opakujeme to, hlavně u pozdějších oddílů, že program je dobré zkracovat, tedy mít ho co nejkratší, pak také snáze žáci obhájí právě své řešení úlohy.

K výuce programování v aplikaci EasyLogo je vhodné používat již mnohokrát zmiňovanou učebnici Informatika pro 1. stupeň (Vaníček, 2012). Ve čtvrté kapitole nazvané „Programování“ se objevuje právě EL a je zde doplněno o různé aktivity. Učebnice slouží i jako opora pro učitele ve výuce.

8. Závěr

EasyLogo, jak už bylo řečeno, je jednoduché programovací prostředí pro děti, které obsahuje výukovou sadu úloh.

Za pomoci dalších učitelů jsem mohla na základě pozorování výuky obnovit a obohatit tuto sadu o své vlastní úlohy.

Touto prací jsem tedy vytvořila novou sadu obsahující 52 úloh. Úlohy jsou rozděleny do sedmi oddílů podle vyučovaných kroků. Seznam je součástí této práce jako příloha. Tato sada byla ozkoušena na základních školách, kde se žákům líbila a učitele zaujala.

Nakladatel učebnice Informatika pro 1. stupeň základní školy (Vaníček, 2012) nahradil původní sadu nově vytvořenou sadou výukových úloh a zveřejnil ji na webových stránkách nakladatelství Computer Press, jako přílohu zmiňované učebnice. Je zde ke stažení celý zazipovaný soubor, který obsahuje novou sadu úloh, seznam aktivit, autorský program GEN pro tvorbu úloh a jeho návod k použití, který jsem sepsala (též přílohou této práce). Tento soubor je v kapitole 4 Programování.

EasyLogo je vhodné k výuce programování na 1. stupni základní školy už pro svou jednoduchost v ovládní a díky svým srozumitelným zadáním úloh. Je dostupné online, jednoduché na spuštění a není potřeba žádné instalace. Po stažení aplikace již není nutné připojení k internetu. EasyLogo je zároveň příjemné pro pedagogy, kteří by si chtěli přizpůsobit úlohy své praxi. Je možné stávající úlohy upravovat, vymazávat, ale mohou i přidávat nové úlohy.

Vzhledem k plánované revizi RVP se domnívám, že takovýchto materiálů, jako je výuková aplikace EasyLogo, bude potřeba čím dál tím více. Na mé práci se mohou podílet další a další učitelé či studenti, kteří budou chtít zlepšovat výuku programování v mladším školním věku.

Zkrácenou verzi ve formě článku jsem publikovala v e-sborníku celostátní konference Počítač ve škole 2017 v Novém Městě na Moravě, které jsem se zúčastnila díky grantu GAJU. Článek je přílohou této práce.

Pevně věřím, že má práce bude přínosem pro učitele prvního stupně základních škol, kteří rádi zařadí programování do své výuky informatiky. Sama tuto aplikaci budu nadále využívat ve své budoucí praxi. Učím na málotřídní škole, kde jedním

z mých předmětů je právě informatika. EasyLogo mám zařazené v tematickém plánu pro školní rok 2016/2017 a i když od následujícího školního roku nebudu působit na této škole, aplikaci si ráda zařadím do tematického plánu i na svém dalším pracovišti. EasyLogo si mě získalo svou jednoduchostí a hravostí.

Literatura

BRDIČKA, Bořivoj. *Informatické myšlení jako výukový cíl*. [online] 22. 4. 2014 [cit. 12. 4. 2017]. Dostupné z WWW: <http://spomocnik.rvp.cz/clanek/18689/INFORMATICKE-MYSLENI-JAKO-VYUKOVY-CIL.html>

ISTE. *Operational Definition of Computational Thinking for K–12 Education*. 2011. [cit. 2014-4-14]. Dostupný z WWW: [<http://www.iste.org/docs/ct-documents/computationa...>].

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY. *Rámcový vzdělávací program pro základní vzdělávání*. [online] 2013 [cit. 12. 3. 2017], Praha: Ministerstvo školství, mládeže a tělovýchovy. Dostupné z WWW: <http://www.msmt.cz/vzdelavani/zakladni-vzdelavani/upraveny-ramcovy-vzdelavaci-program-pro-zakladni-vzdelavani>

PECINOVSKÝ, Rudolf. *Baltík: učebnice programování nejen pro děti*. S.l.: SGP Systems, 2000.

PECINOVSKÝ, Rudolf a Miroslav VIRIUS. *Učebnice programování: základy algoritmizace : učebnice s příklady v Turbo Pascalu a Borland C++*. Praha: Grada, 1997. ISBN 80-7169-577-7.

SALANCI, Ľubomír. *EasyLogo-discovering basic programming concepts in a constructive manner*. In: *Constructionist approaches to creative learning, thinking and education: Lessons for the 21st century*. Bratislava : FMFI UK, 2010. ISBN 978-80-89186-66-2, ISBN 978-80-89186-65-5.

SCHÖN, Otakar. *Programovat může každý, stačí chtít a být zvědavý*. Hospodářské noviny, 2016

STÁRKOVÁ, Dagmar. *Model ADDIE při vytváření koncepce výuky a jeho aplikace*. [online] 2012 [cit. 25. 3. 2017], Pedagogická fakulta Univerzity Karlovy. Dostupné z WWW: http://it.pedf.cuni.cz/strstud/edutech/2012_Addie_Starkova/

THE LEAD PROJECT. *Programování pro děti: naučte se programovat při tvorbě skvělých her*. Brno: Computer Press, 2013. ISBN 978-80-251-3809-0.

VANÍČEK, Jiří. *Informatika pro 1. stupeň základní školy: informační a komunikační technologie*. V Brně: Computer Press, 2012. ISBN 978-80-251-3749-9.

SCRATCH [online] Lifelong Kindergarten Group v MIT Media Lab. Dostupné z: <https://scratch.mit.edu/>

Publikace

ROSECKÝ, Jan. *Počítač ve škole 2017 – sborník příspěvků*. Nové Město na Moravě: Gymnázium Vincence Makovského se sportovními třídami, 2017. ISBN: 978-80-905765-8-2

Přílohy

Příloha č. 1: CD-ROM

- Diplomová práce ve formátu PDF
- Aplikace EasyLogo se všemi dokumenty zkomprimovaná do jednoho archivu zip

Příloha č. 2: Seznam úloh rozdělených do oddílů

Seznam nových aktivit

Mód přímý

1. Úvod, instrukce
2. Auto do cíle L
3. Auto do cíle dvojitě L
4. Auto do cíle U
5. Auto do cíle křivka
6. Auto do cíle klikatice
7. Přeškrtnání písmen – libovolné
8. Přeškrtnání písmen – v pořadí
9. Přelet na květinu, vylet z okna

Tvorba programu

10. Auto do cíle – L
11. Navleč korálky – libovolné
12. Navleč korálky – určené pořadí
13. Golem chodba
14. Gumování jablíček čarou
15. Gumování jablíček puntíkem
16. Gumování jablíček puntíkem
17. Písmeno T
18. Trojúhelník
19. Traktor
20. Letadlo
21. Anička skáče panáka

Čárkovaná čára

22. Nakresli čárkovanou čáru
23. Rozdělení bytu
24. Spojení ostrovů

Opakuj

25. Objevení čtverce – včelka
26. Krtek kreslí čtverec
27. Značka stop
28. Včelka a květiny
29. Anička skáče panáka
30. Traktor – pluh

Řešíme problémy

31. Umístí písmeno
32. Vzpěrač – činka
33. Stromy
34. Žralok
35. Ornamenty

Opravujeme chyby

36. Písmeno D
37. Písmeno D – 2. varianta
38. Písmeno D – 3. varianta
39. Robot – chyba programu
40. Cesta na stadion
41. Jiný cíl cesty (budovy)
42. Letadlo

Procedury

43. Praporek – čtverec
44. Větrník
45. Ohrádka pro zvířátka
46. Domek
47. Panelák
48. Ulice
49. Náměstí
50. Osmička
51. Robot – ornament
52. Volný režim

Příloha č. 3: Návod na práci s autorským programem GEN

Autorský program GEN pro tvorbu úloh v EasyLogo

Autorský program GEN pro tvorbu úloh vytvořil student Jihočeské Univerzity Stanislav Hupka.

Pro spuštění tohoto programu je potřeba mít v PC nainstalovaný program Java.

Pomocí tohoto programu lze vytvořit novou úlohu bez složitého vypisování programu do Poznámkového bloku. V programu se napíše zadání úlohy, nahraje se příslušné pozadí, tvar, který se bude pohybovat a musí se vybrat i mód ve kterém se bude úloha řešit.

Módy:

- normální mód (simple), kde se přetahují příkazy z postranní lišty
- přímý mód (direct) zde se tvar želvy ovládá pomocí tlačítek pohybu
- mód s možností vkládání procedur (procedural)

Zároveň se v programu navolí i velikost mřížky a startovní souřadnice pohybujícího se tvaru. Souřadnice se určí pouze kliknutím do prostoru, kde chceme, aby měl pohybující se tvar, vůči pozadí, startovní pozici. Pomocí šipek si zvolíme natočení tvaru.

Takto připravená úloha se jen potvrdí a program vygeneruje textový soubor. Text se musí uložit do stejné složky, která obsahuje jak příslušné pozadí, tak i tvar zmíněný v naprogramovaném souboru. Po spuštění programu EL je pak tato úloha zcela normálně funkční.

Výuka programování na 1. stupni ZŠ v prostředí EasyLogo

Martina Bartíková

e-mail: bartikovamartina2@seznam.cz

Jihočeská univerzita v Českých Budějovicích, Pedagogická fakulta

Úvod

Pro výuku programování na 1. stupni existuje celá řada programů a prostředí. Tato práce je však zaměřena na program EasyLogo, na znění a posloupnost jeho úloh. Původní sadu úloh tohoto programu jsem upravila, vytvořila nové úlohy a poskládala je v prostředí tak, aby jejich výuka byla snadná, smysluplná a žáci jí dobře rozuměli.

Programování na 1. stupni ZŠ

Do výuky informatiky se programování v rámci České republiky zařazuje velmi zřídka, natož pak právě na prvních stupních. Předmět Informatiky je do vyučovacího plánu zařazen ve čtvrtém, někdy až v pátém, ročníku a pokud jej vyučují sami učitelé prvního stupně, nikoli učitel z druhého stupně zaměřený na informatiku, jejich zkušenosti bývají často velmi omezené. I proto se do výuky zařazují pouze základy práce s textovými editory, vytváření složky, kopírování, práce s internetem a někdy i práce v MS PowerPoint. Programování tedy na prvním stupni učitelé do vyučovacího plánu nezařazují, a to bych právě se svou prací chtěla změnit. Vybrala jsem jednoduché programovací prostředí EasyLogo, které je nenáročné jak pro učitele, tak pro samotné žáky. V počítači není potřeba žádná instalace, po stažení programu stačí pouze rozbalit soubor a pak jej jednoduše spustit. Výuku v tomto programu jsem sama odučila, žákům se práce s tímto programem velice líbila a bylo to pro ně zpestření výuky pomocí zábavné a hravé formy.

O programu EasyLogo

„EasyLogo bylo vyvinuto pro lidi se základními počítačovými znalostmi, aby pro ně bylo programování a řešení problémů co možná nejjednodušší.“ (L. Salanci)

Jedná se o jednoduché programovací free prostředí, pro pohyb je zde použita mřížka (čtvercová síť) a ovládá se zvolený tvar (auto, želva,..). Autorem tohoto programu je doc. RNDr. Lubomír Salanci, PhD. V programu se pracuje ve dvou režimech – Řešení úloh a Vlastní tvorba.

Úlohy v tomto programu jsou již vytvořené, žáci plní zadání v režimu „Řešení úloh“. Cílem je naučit žáky programování od jednoduchých funkcí až po ty složitější a podle toho jsou řazeny i jednotlivé, výše zmiňované oddíly a jejich úlohy. Svou práci zde má i učitel, program sám nevyhodnocuje správnost řešení, tudíž je tato role právě na něm.

Po přepnutí do režimu „Vlastní tvorba“ mohou žáci vytvářet svůj vlastní program a rozvíjet tak svou kreativitu a logické myšlení.

Program je dle mého názoru velmi jednoduchý na pochopení, je hezky zpracovaný a žáci zde mohou rozvíjet svou kreativitu a logické myšlení při řešení úloh. Sama jsem si ho velmi rychle oblíbila a je používaným prvkem v mé praxi, kdy v něm mí žáci velice rádi pracují. Úlohy ve většině případů mohou mít i několik možností, které lze se žáky diskutovat, odůvodňovat a rozebírat, žáci se tak mohou naučit obhájit si svou práci.

1.1 Výhody

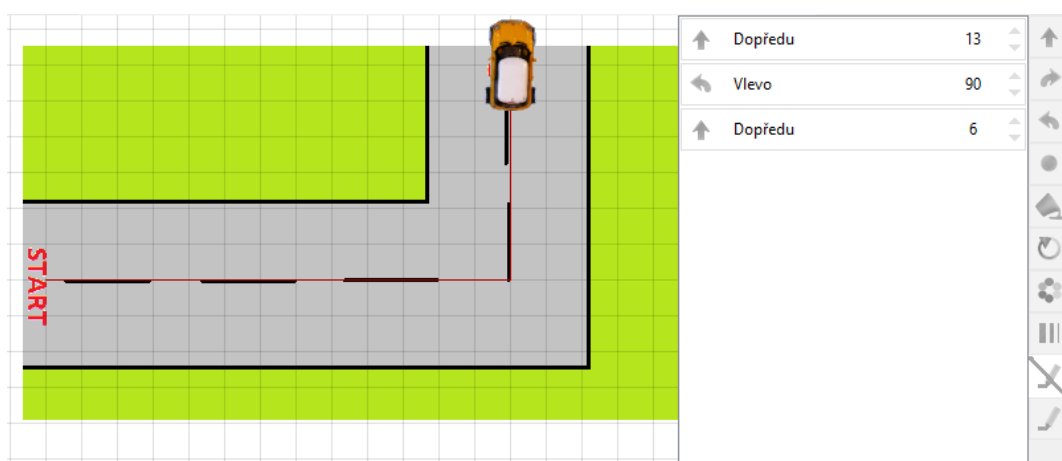
- Program nevyžaduje instalaci
- Jedná se o jednoduché prostředí
- Objekt se pohybuje v rámci mřížky
- Ihned po zadání příkazu vidíme jeho provedení
- Uzavřené pedagogické prostředí

1.2 Nevýhody

- Nelze zde vytvořit zaoblené tvary či kruh
- Program sám nevyhodnocuje správnost řešení
- Program není dostupný v online verzi

Jde o soubor řady úloh, které postupně rozvíjejí dovednosti žáka. Pracuje se zde s pohybujícím se tvarem např. želvy či auta, který se pohybuje v prostoru se

čtvercovou sítí, kde jsou jasně viditelné kroky příkazů. Jeden krok znamená posunutí se o jeden čtverec. K pohybu se využívají zprvu směrové šipky, následně pak příkazy (dopředu, vpravo, vlevo, opakuj, kresli/nekresli, udělej puntík,..), které je nutno přesunout do příkazového sloupce a poskládat je za sebe do správného pořadí, v jakém mají být vykonány. U jednotlivých příkazů je možné pozměnit parametr, aby se za sebe neskládaly několikrát stejné příkazy, např. dopředu. Na následujícím obrázku je vpravo vidět nástrojová lišta s jednotlivými příkazy, hned vedle je pracovní plocha s vytvořeným programem pro pohyb auta. U příkazů jsou vidět jejich parametry.



Obr. 1 - Ukázka prostředí programu EasyLogo – příklad řešení úlohy

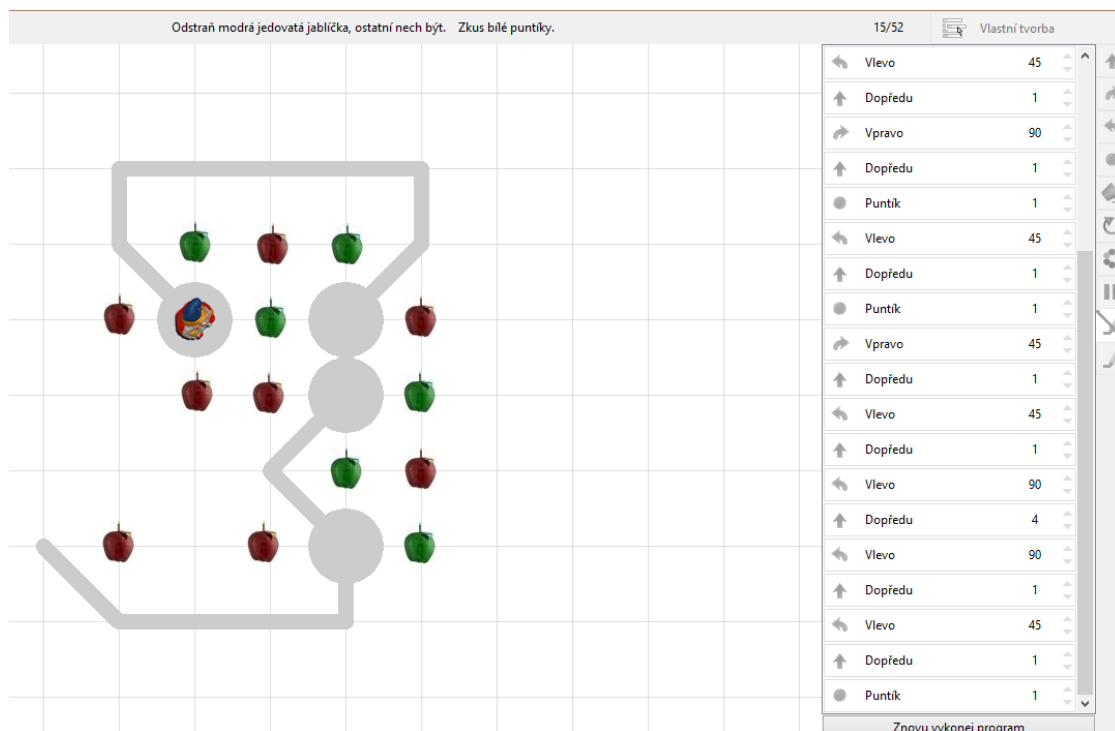
Výukový plán této aplikace je rozdělen do sedmi oddílů, které postupně vyučují základy programování a zabývají se vždy zlomovým příkazem či tématem. Jako první je v sadě zařazen oddíl nazvaný „Mód přímý“, který obsahuje devět úloh. Tyto úlohy jsou ovládány pomocí směrových šipek a slouží k seznámení žáků se samotnou aplikací. Následuje oddíl „Tvorba programu“ v němž je jedenáct úloh. Zde žáci tvoří samotný program pro pohybující se objekt. Skládají dané příkazy (dopředu, vpravo, vlevo, kresli/nekresli, opakuj, udělej puntík) do příkazového rámečku za sebe a objekt vykonává jejich pokyny. Další oddíly jsou věnovány složitějším případům v programu, jako je tvorba čárkované čáry, tudíž příkaz „kresli/nekresli“, dále pak příkaz „opakuj“, do kterého je nutné vnořit další příkazy z nabídky. A následující oddíly využívají již předem hotové tvary, např. čtverec, a nebo oddíl, ve kterém se žáci učí hledat chybu v programu a opravit ji.

Vytváření nové sady úloh

EasyLogo jsem si vybrala pro svou diplomovou práci, kdy jsem se rozhodla upravit a rozšířit původní sadu úloh tohoto program. Celá práce byla uskutečněna ve čtyřech fázích:

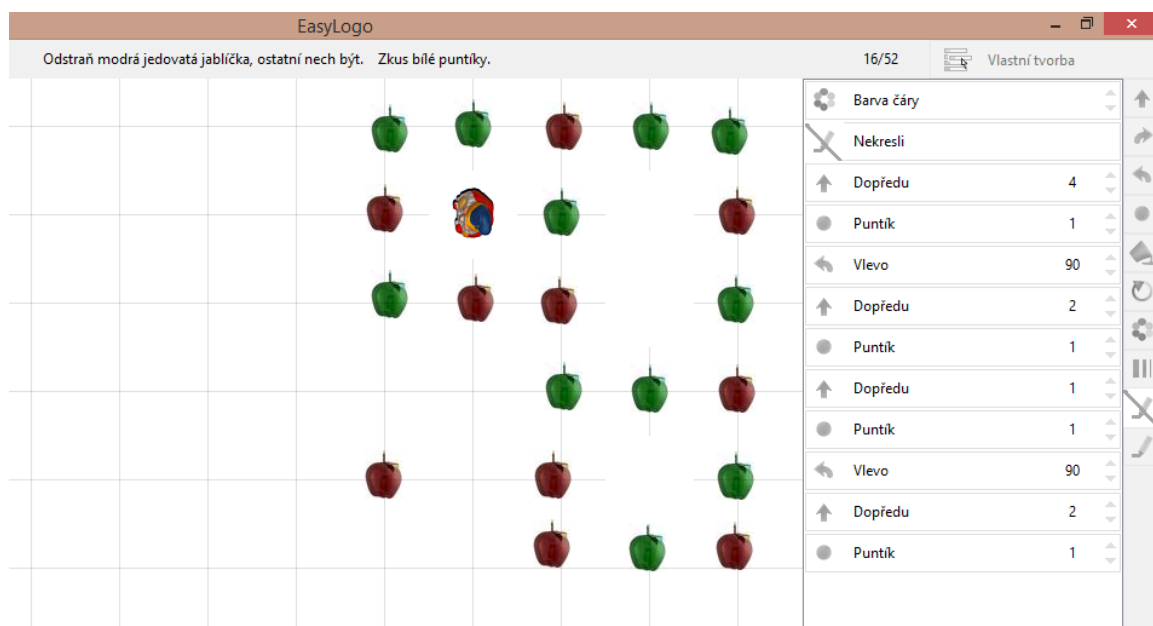
1. Analýza původních úloh (pozorování výuky)
2. Rozbor poznatků
3. Vytváření nových úloh
4. Ověření nové sady ve výuce

V první fázi mé práce jsem testovala původní sadu na řadě základních škol a z pozorování vyplývá, že tato sada byla pro žáky v některých případech moc rychlá a někdy nejasná. U těžších příkazů byl postup uspěchaný, u lehčích bylo naopak úloh více. Tyto „nedostatky“ jsem se pokusila odstranit, ve složitějších případech úlohy přidat, v některých případech stačilo jen upravit zadání nebo pozadí úlohy. Nová sada tedy obsahuje úlohy původní, upravené a úlohy zcela nové. Pro příklad jsem vybrala upravenou úlohu, kterou můžeme vidět na obrázku č. 2.



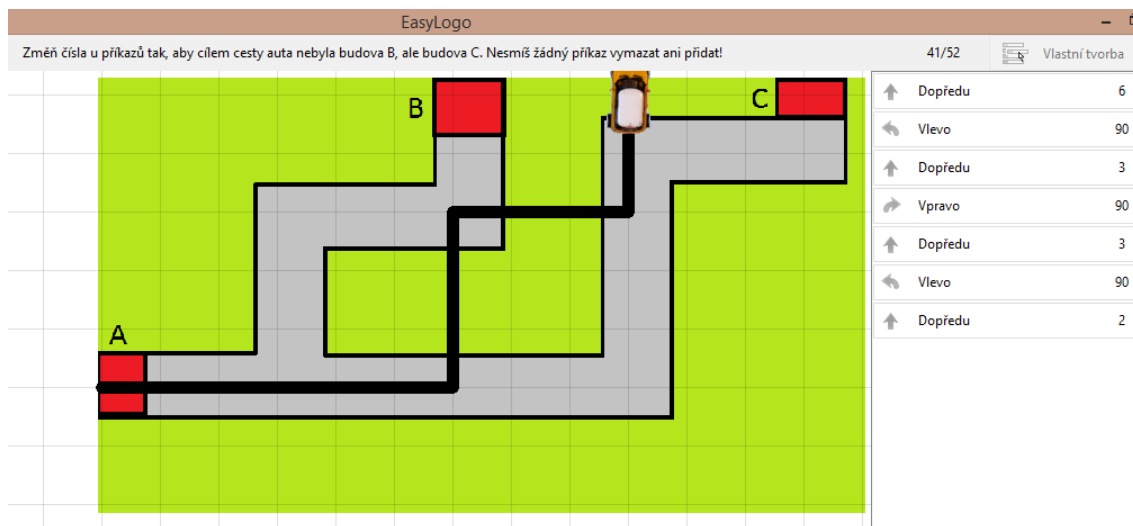
Obr. 2 – Původní úloha, zadání: Odstraň modrá jedovatá jablčka, ostatní nech být. Zkus bílé puntíky. – Nesprávné řešení.

Na obrázku č. 2 vidíme původní úlohu, kterou žák vyřešil tak, že kličkoval pomocí úhlopříček mezi jablky, přičemž ale příkazem, který by se měl v úloze použít, je příkaz „kresli/nekresli“. Šedá barva dráhy je zde použita záměrně, aby byl patrný pohyb tvaru (trpaslíka) po úhlopříčkách mřížky.



Obr. 3 – Upravená úloha – správné řešení.

Na obrázku č. 3 má žák stejný úkol jako na přechozím obrázku, nicméně je zde pozměněné pozadí, kam jsem přidala jablka, abych tak zamezila pohybu po úhlopříčkách. Nyní musí žák využít příkazů „kresli/nekresli“, které chceme v úloze naučit. Jedná se o jednu z upravených úloh.



Obr. 4 – Nově vytvořená úloha., zadání: Změň parametry u příkazů tak, aby cílem cesty auta nebyla budova B, ale budova C. Nesmíš žádný příkaz vymazat ani přidat!

Na obrázku č. 4 vidíme jednu nově vytvořenou úlohu. Úkolem je změnit cíl cesty. Žáci se zde učí hledat a opravovat chyby v programu.

Tyto upravené či nově vytvořené úlohy jsem znovu vyzkoušela ve výuce Informatiky na 1. stupni základních škol a následně jsem o této práci hovořila s učiteli i se žáky.

Výsledky ověřování experimentální výuky

Při testování na základních školách, ať už to bylo testování původní sady, tak i nové sady úloh, jsem slyšela od učitelů, kteří program dříve neznali, mnoho kladných ohlasů. EasyLogo pro ně bylo velmi příjemným zpestřením hodin informatiky. Žáci se naučili něco nového a to pomocí zábavné formy, mohli mezi sebou soutěžit a to v nich jen podporovalo zájem o tento program.

Pro učitele bylo mnohdy seznámení s tímto programem podnětem pro zařazení výuky programování do svých tematických plánů. Byli vděční za ukázkou tohoto programu a také za rozšířenou sadu úloh, kterou budou moci ve své výuce použít.

Žákům se tento program většinou velice líbil. Zvláště pak úlohy, kde mohli mezi sebou soupeřit o to, kdo vymyslí program s co nejmenším počtem příkazů, jak je tomu např. v úloze č. 21 (v nové sadě), zde mají žáci za úkol doletět s letadlem z Prahy do Madridu, nebo u úlohy č. 30 (v nové sadě), kdy mají za úkol pomoci farmáři doorat jeho pole.

Závěr

V mé práci se mi podařilo nejen rozšířit a upravit sadu úloh výukové aplikace EasyLogo, ale i dostat tuto aplikaci do podvědomí řadě učitelů na 1. stupni. Doufám, že se pohled na programování na základních školách změní k lepšímu a že jeho vyučování už nebude zamítáno, jako tomu bylo dříve. Programování totiž žákům pomáhá rozvíjet jejich kreativitu a logické myšlení a pomocí takovýchto aplikací může být jeho výuka i zábavou.

Dostupnost výukové aplikace

Výuková aplikace s upravenou sadou úloh je dostupná na webových stránkách nakladatelství Computer Press jako příloha učebnice Informatika pro 1. stupeň základní školy (Vaniček, 2012), v kapitole 4 Programování.

<http://www.knihy.cpress.cz/informatika-pro-1-stupen-zakladni-skoly.html>

Poznámka

Výzkum byl podpořen z grantového projektu GAJU 121/2016/S.

Literatura

VANIČEK, Jiří. *Informatika pro 1. stupeň základní školy: informační a komunikační technologie*. V Brně: Computer Press, 2012. ISBN 978-80-251-3749-9.

Salanci, Lubomír: EasyLogo-discovering basic programming concepts in a constructive manner. In: Constructionist approaches to creative learning, thinking and education: Lessons for the 21st century. Bratislava : FMFI UK, 2010. ISBN 978-80-89186-66-2, ISBN 978-80-89186-65-5.