

**UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY**

**ABSOLÚTNY A RELATÍVNY POHYB VO VYUČOVANÍ  
PROGRAMOVANIA**

Záverečná práca dopĺňujúceho pedagogického štúdia

**UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY**

**ABSOLÚTNY A RELATÍVNY POHYB VO VYUČOVANÍ  
PROGRAMOVANIA**

Záverečná práca doplnujúceho pedagogického štúdia

Študijný program:	Informatika
Študijný odbor:	7656 Učiteľstvo akademických predmetov
Školiace pracovisko:	Katedra základov a vyučovania informatiky
Školiteľ:	doc. PaedDr. Monika Tomcsányiová, PhD.

**Bratislava, 2016**

**Bc. Lucia Budinská**



## **Čestné vyhlásenie**

Čestne vyhlasujem, že som záverečnú prácu „Absolútny a relatívny pohyb vo vyučovaní programovania" vypracovala samostatne s použitím uvedených zdrojov.

V Bratislave dňa 31.05.2016

---

Bc. Lucia Budinská

## **Pod'akovanie**

Rada by som sa poďakovala všetkým, ktorí sa nejakým spôsobom podieľali na vzniku tejto práce. Svojej školiteľke doc. PaedDr. Monike Tomcsányiovej, PhD. ďakujem za cenné rady, nápady a pripomienky k práci. Simone Vajdiarovej, jej žiakom a svojim žiakom z Freedu ďakujem za pomoc pri prvom testovaní. Pani učiteľke Mgr. Martine Prokajovej a jej žiakom z 3.B na ZŠ J. Horáka ďakujem, že si našli čas a otestovali úlohy v prostredí GEVIN. A v neposlednom rade ďakujem Petrovi Šulíkovi za grafický návrh kartičiek.

## Abstrakt

Závěrečná práce k doplňujícímu pedagogickému štúdiu z informatiky sa zaoberá vyučováním informatickej výchovy respektíve informatiky na prvom stupni základných škôl. Venujeme sa okruhu zameranému na algoritmizáciu a programovanie a skúmame dva najčastejšie používané typy pohybu postavičky, vykonávateľa programu, a to absolútny a relatívny pohyb. Bližšie opisujeme používané detské programovacie prostredia a edukačné hry, analyzujeme niekoľko takto zameraných úloh z informatickej súťaže iBobor a následne prezentujeme výsledky vlastného výskumu. Jeho prvá časť prebiehala so žiakmi piatych ročníkov a zistili sme, že títo žiaci viac preferujú absolútny pohyb a robia v ňom menej chýb. Následne sme navrhli sadu šestnástich úloh v edukačnom programe GEVIN, ktorá mala žiakov postupne zoznámiť s oba typmi pohybov. Otestovali sme ju so žiakmi tretieho ročníka a na základe zistených problémov navrhujeme odporúčania pre prax.

**Kľúčové slová:** programovanie, absolútny pohyb, relatívny pohyb, informatika, prvý stupeň základnej školy, vyučovanie

## **Abstract**

This diploma thesis to complementary pedagogical education in informatics is aimed at informatics education in lower levels of primary schools. We are interested in computing and we study two most common motions of figure (as program interpreter) – absolute and relative motion. We describe some of the commonly used children programming languages and educational games, we also analyse a few motion based tasks from informatics competition iBobor. Afterwards we present results of our own research. Its first part took place with fifth grade pupils and we found out that they prefer absolute motion and they make less mistakes in this type of motion. Next we designed set of sixteen tasks in educational program GEVIN which was aimed to acquaint pupils with both of the motions. We tested this set with third grade pupils and from the conclusions we offer recommendation to educational practice.

**Keywords:** computing, absolute motion, relative motion, informatics, primary schools, education

# Obsah

Úvod.....	1
1. Teoretické východiská .....	2
1.1. Informatika v Štátnom vzdelávacom programe.....	2
1.2. Piagetova teória kognitívneho vývinu .....	4
1.3. Bloomova taxonómia vzdelávacích cieľov.....	4
1.4. Programovacie prostredia .....	5
1.5. Hry na rozvoj algoritmickeho myslenia.....	10
2. Absolútny a relatívny pohyb .....	14
2.1. iBobor .....	14
2.2. Testovanie motivácie .....	19
2.3. GEVIN .....	22
3. Testovanie .....	25
3.1. Návrh úloh .....	25
3.2. Priebeh a výsledky testovania.....	27
3.3. Návrhy na zlepšenie.....	30
Záver .....	31
Bibliografia .....	32
Prílohy.....	34



# Úvod

Na prvom stupni slovenských základných škôl sa od školského roku 2011/12 vyučuje predmet informatická výchova, ktorý bol v inovovanom Štátnom vzdelávacom programe z roku 2014 premenovaný na predmet informatika. Jedným z tematických okruhov, ktorým sa majú žiaci venovať, je aj algoritmizácia a programovanie. Existuje viacero prístupov k vyučovaniu tohto okruhu, my sme si vybrali jeden z najrozšírenejších a to ovládanie postavičky. Zameriavame sa hlavne na rozdiely v pohybe ovládanej postavičky, ktorý môže byť absolútny alebo relatívny, a rozdiely v náročnosti chápania tohto pohybu pre žiakov. Cieľom našej práce je odhaliť, aké problémy môžu mať žiaci pri učení sa týchto konceptov a navrhnúť vhodné úlohy, ktoré by im ich pomohli odstrániť.

Prvá kapitola sa bude venovať teoretickým východiskám – štátnemu vzdelávaciemu programu pre predmet informatická výchova, resp. informatika, Bloomovej taxonómii vzdelávacích cieľov a taktiež opíšeme časť Piagetovej teórie kognitívneho vývoja, ktorá sa dotýka žiakov prvého stupňa základných škôl. Súčasťou prvej kapitoly bude aj opis vybraných malých programovacích jazykov a edukačných hier využívajúcich ovládanie postavičky na učenie algoritmizácie a programovania.

V druhej kapitole sa sústredíme na absolútny a relatívny pohyb v programovaní. Zanalyzujeme niekoľko úloh zo súťaže iBobor a opíšeme výsledky nášho prvotného testovania úloh so žiakmi 5. ročníka základných škôl, ktorého cieľom bude zistiť vhodnosť nami použitej motivácie a taktiež odhaliť najväčšie problémy, ktoré žiaci s uvedenými typmi pohybu majú. Tieto zistenia budú slúžiť na návrh úloh v prostredí GEVIN, ktoré v tejto kapitole predstavíme.

Poslednú kapitolu venujeme testovaniu finálnej sady úloh v prostredí GEVIN so žiakmi prvého stupňa. Bližšie opíšeme postupnosť jednotlivých úloh a priebeh testovania. Vyhodnotíme úspešnosť žiackych riešení, ale zameriame sa aj na rôzne postupy žiackej práce a z nich vyplývajúce výsledky. Na ich základe navrhne zlepšenia softvéru a odporúčania pre pedagogickú prax, ktoré budú tiež súčasťou tejto kapitoly.

# 1. Teoretické východiská

V tejto kapitole sa budeme venovať vyučovaniu informatiky, a to hlavne oblasti zaoberajúcej sa algoritmicizáciou a programovaním, na prvom stupni základných škôl na Slovensku. Okrem vedomostí, ktoré majú žiaci na prvom stupni získať, nás bude zaujímať aj v akom vývinovom štádiu sa z hľadiska Piagetovej teórie nachádzajú a aké obmedzenia to môže priniesť do vyučovania informatiky. Taktiež v tejto kapitole bližšie opíšeme vybrané programovacie prostredia a edukačné aktivity, v ktorých sa využíva pohyb robota alebo postavičky, na základe ktorých budeme navrhovať vlastné úlohy.

## 1.1. Informatika v Štátnom vzdelávacom programe

Predtým, ako začneme uvažovať nad vhodnosťou aktivít pre žiakov, musíme vedieť, aké zručnosti majú v danom predmete získať. Tie na Slovensku určuje Štátny vzdelávací program (ďalej len ŠVP) [1], ktorý vstúpil do platnosti v školskom roku 2011/2012. V aktuálnom školskom roku sa podľa neho učia žiaci druhého až štvrtého ročníka. Je rozdelený do 4 tematických oblastí:

- 1) Informácie okolo nás
- 2) Komunikácia prostredníctvom IKT
- 3) Postupy, riešenie problémov, algoritmicke myslenie
- 4) Princípy fungovania IKT
- 5) Informačná spoločnosť

Tretia oblasť, *Postupy, riešenie problémov, algoritmicke myslenie*, sa venuje algoritmicizácii a základom programovania. Dôležitými pojmami, kľúčovými pre tento okruh, sú, okrem iných, *riadenie robota, postupnosť krokov, detský programovací jazyk, program*. ŠVP ďalej uvádza, aké postupy a metódy na vyučovaní použiť. Žiaci majú v počítačovom prostredí riešiť úlohy pomocou robota, skladať obrázky z menších obrazcov, používať aj okamžité vykonávanie príkazov, aj vykonávanie pripravenej postupnosti. Taktiež sa majú naučiť riešiť jednoduché algoritmy v detskom programovacom prostredí, ako je napr. kreslenie obrázkov alebo pohyb animovaných obrázkov [1].

Žiak by mal mať po absolvovaní prvého stupňa základnej školy základy algoritmickeho myslenia, poznať detské programovacie prostredia a riešiť úlohy v nich pomocou priameho režimu.

Priamo z postupov a metód vyplýva, že vhodnými úlohami pre žiakov tohto stupňa sú úlohy, v ktorých ovládajú robota (alebo inú postavičku), a to v dvoch režimoch – s okamžitým vykonávaním príkazov a vykonávania pripravenej postupnosti krokov. ŠVP priamo neurčuje a ani neodporúča edukačné prostredie, či programovací mini jazyk, ktoré majú byť použité na vyučovanie.

Predmet Informatická výchova má časovú dotáciu 3 hodiny [2] za celý prvý stupeň, takže na väčšine škôl sa učil v druhom až štvrtom ročníku po jednej hodine týždenne.

V školskom roku 2015/2016 vstúpil do platnosti Inovovaný štátny vzdelávací program (ďalej iŠVP) [3], v ktorom bol predmet informatická výchova premenovaný na informatika, čím sa zjednotil názov na všetkých troch stupňoch vzdelávania. Okrem zmeny názvu nastáva v iŠVP aj zmena časovej dotácie – predmet informatika je určený pre žiakov tretieho a štvrtého ročníka a to s rovnakou dotáciou – 1 hodinou týždenne [4]. (Prví žiaci s informatikou podľa iŠVP tak budú tretiaci v roku 2017/2018.)

iŠVP má rovnako 5 oblastí, zmenili sa len názvy:

- 1) Reprezentácia a nástroje
- 2) Komunikácia a spolupráca
- 3) Algoritmické riešenie problémov
- 4) Softvér a hardvér
- 5) Informačná spoločnosť

Oblasť Algoritmické riešenie problémov je následne rozdelená na päť častí:

- a) Analýza problému
- b) Interaktívne zostavovanie riešenia
- c) Riešenie problémov pomocou postupnosti príkazov
- d) Interpretácia zápisu riešenia
- e) Hľadanie a opravovanie chýb

Výkonové štandardy zostali viac menej podobné tým v ŠVP, mierne sa upravila iba ich forma. Rovnako ako ŠVP aj iŠVP hovorí o dvoch režimoch ovládania postavičky (v iŠVP označenej ako „vykonávateľ“) – okamžité (alebo interaktívne) zadávanie krokov a vykonávanie postupnosti príkazov. V iŠVP pribúda aj interpretácia zápisu riešenie

a hľadanie a opravovanie chýb, ktoré explicitne neboli spomínané v ŠVP, ale pri praktickej práci na programoch či postupnostiach krokov sa s nimi určite žiaci stretli a teda sa ich učili.

iŠVP tak zdôrazňuje potrebu žiakov vedieť krokovať svoje programy, interpretovať ich, hľadať v nich chyby a diskutovať o riešeniach a možnostiach zlepšenia.

## **1.2. Piagetova teória kognitívneho vývinu**

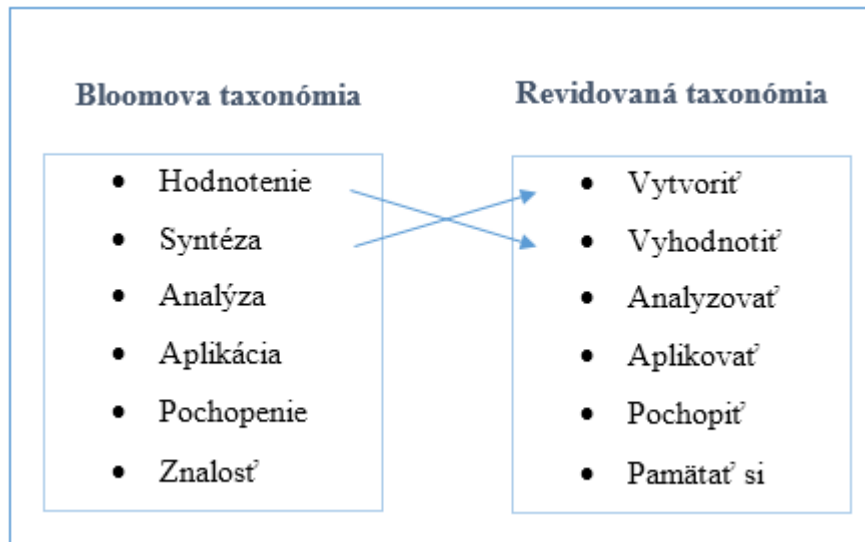
Pri vyučovaní žiakov musíme brať do úvahy aj to, v ktorom vývinom štádiu sa nachádzajú – pri informatike je dôležité hlavne kognitívne hľadisko vývinu. Kognitívny vývin najlepšie opísal Piaget, ktorý zdefinoval štyri štádiá – sezomotorické, predoperačné, štádium konkrétnych operácií a štádium formálnych operácií [5]. Žiaci prvého stupňa základných škôl, t.j. vo veku cca 6 – 11 rokov, sa nachádzajú v štádiu konkrétnych operácií.

Deti v tomto štádiu už dokážu logicky uvažovať, pomocou indukcie generalizujú a na veci či situácie sa vedia pozeráť aj z pohľadu niekoho iného. V tomto štádiu však ešte nie je úplne rozvinuté abstraktné myslenie, deti majú problémy s hypotetickým uvažovaním a dedukovaním. Väčšina ich učenia je v štýle pokus - omyl, čo sa odráža aj na ich logike. Ťažkosti s dedukovaním budúcich udalostí na základe generalizovaných princípov a nedostatočná schopnosť hypoteticky myslieť môže vytvárať prekážky pri vyučovaní programovania.

Učiteľia by tak mali dať žiakom dostatok priestoru na veľa pokusov, pri zoznamovaní sa s prostredím a programovaním, aby neskôr žiaci dokázali generalizovať správanie programu, resp. postavičky. Namiesto hypotetických situácií ich nechať priamo, konkrétne, modelovať a namiesto abstraktných pojmov či princípov využívať reprezentácie, ktorým žiaci porozumejú.

## **1.3. Bloomova taxonómia vzdelávacích cieľov**

Pri stanovovaní vzdelávacích cieľov – či sa jedná o dlhodobé, alebo iba o čiastkové ciele, sa často využíva Bloomova taxonómia. Je vhodná aj pri vytváraní úloh, alebo ich hodnotení. Edukačný psychológ Benjamin Bloom ju vytvoril v roku 1956, no v roku 2000 vznikla takzvaná revidovaná Bloomova taxonómia vzdelávacích cieľov, pod ktorú sa podpísali Lorin Anderson a David Krathwohl [6]. Zmenili poradie cieľov a trochu upravili ich názvy. Na Obr. 1 vidíme porovnanie oboch teórií.



Obr. 1 - Rozdiely v taxonómiách

Jednotlivé stupne revidovanej Bloomovej taxonómie si môžeme bližšie popísať na príklade s programovaním. Na prvú úroveň, *pamätať si*, sa žiaci dostanú vtedy, keď sa naučia jednotlivé programovacie príkazy alebo úkony. *Pochopia* ich vtedy, keď budú vedieť povedať, čo vykonajú a na čo sa používajú. Následne by ich mali vedieť *aplikovať* a teda použiť v nejakom programe. *Analýzou* je vysvetlenie svojho riešenia, alebo určenie, čo spraví zadaný program – musia ho analyzovať. *Hodnotiť* môžu svoje alebo cudzie programy – z hľadiska správnosti, efektívnosti, prehľadnosti a podobne. V poslednom stupni by žiaci mali vedieť *vytvoriť* vlastný projekt, bez zadania – takže použiť všetky predchádzajúce vedomosti, navrhnúť si vlastnú tému, vhodne zvoliť nástroj (jazyk, prostredie), aplikovať podobné postupy, zistiť, čo treba upraviť, zhodnotiť správnosť a funkčnosť a podobne.

Dôležité je, aby žiaci dosiahli vo vyučovacom procese aj k tým najvyšším stupňom vzdelávacích cieľov, keďže tie sú viac motivačné a vyžadujú od nich zapojenie vlastnej tvorivosti. A hlavne, v porovnaní s učením iba vo fáze zapamätávania, takéto učenie je zmysluplné [7].

#### 1.4. Programovacie prostredia

Na vyučovanie programovania žiakov na najnižšom stupni vzdelávania sa využívajú takzvané mini jazyky. Podľa [8] sú to (väčšinou) vizuálne jazyky prispôbené na uvedenie žiakov do sveta programovania. Mali by byť jednoduché a zrozumiteľné, názorné, atraktívne pre žiakov, mali by poskytovať dialógový režim a byť modulárne.

Výhodou použitia mini jazyka na vyučovanie základov programovania je jeho jednoduchá syntax – žiaci sa tak nemusia učiť veľa príkazov, dokonca ak sú obrázkové, či kartičkové, stačí ak porozumejú ich významu. Špeciálne pri kartičkových jazykoch je veľkou výhodou nemožnosť syntaktických chýb, ktoré by žiakov mohli zbytočne rušiť a odpútať ich pozornosť od riešených problémov a aktivít.

Niektoré mini jazyky vznikli ako zjednodušená forma väčších, detských, jazykov – spomenieme napríklad dvojicu Logo a EasyLogo, či dvojicu Scratch a ScratchJr. Takéto vzťahy uľahčujú žiakom neskoršiu prácu s „väčším“ jazykom, keďže základné princípy pochopili na jednoduchšej verzii jazyka.

Mini jazyky sa vyznačujú aj tým, že väčšinou ponúkajú sadu zmysluplných a atraktívnych úloh pre žiaka, ktoré majú rastúcu náročnosť. Toto jednak posilňuje motivačný faktor učenia, a zároveň žiaci postupne prejdú niekoľko štádií Bloomovej taxonómie. Aby však dosiahli to najvyššie – štádium tvorby – mali by mini jazyky ponúkať aj takzvaný voľný mód – v ktorom žiaci môžu vytvárať vlastné programy bez predpripraveného zadania úlohy.

Medzi najznámejšie a najviac používané mini jazyky patria napríklad Baltík, EasyLogo a Scratch Jr. V nasledujúcej časti si ich bližšie opíšeme.

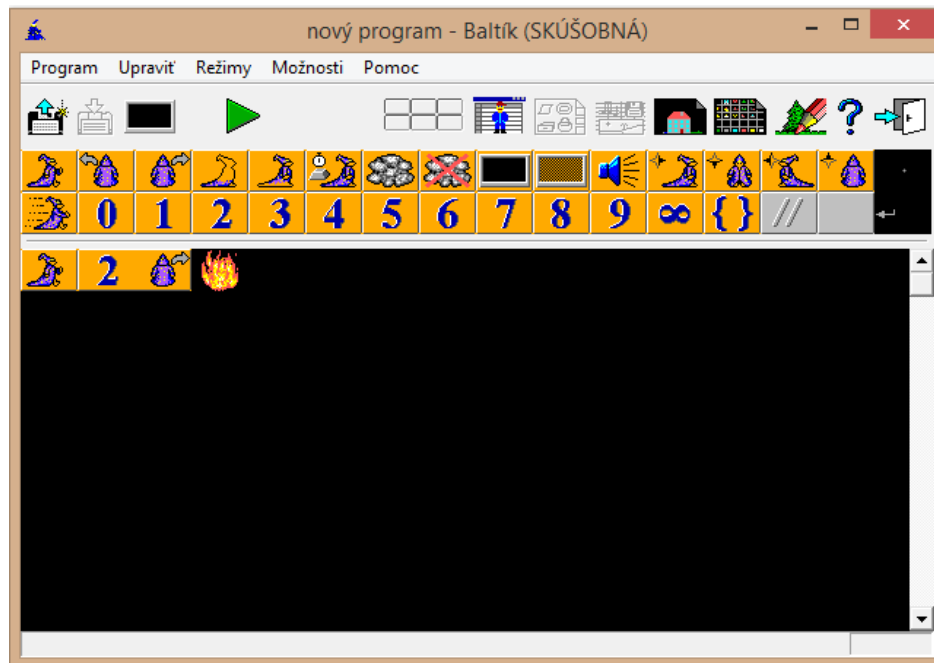
#### **1.4.1. Baltík**

Baltík [9] je detský programovací jazyk, ktorý obsahuje niekoľko režimov. Prvý, „Skladať scénu“, je určený na zoznámenie s prostredím a žiaci v ňom pomocou myši skladajú scénu do štvorcovej mriežky (ktorá síce nie je viditeľná, ale štvorceky z obrázkami do nej automaticky zapadajú). Tento režim nie je programovací.

Druhý režim je už prípravou na programovanie, volá sa „Čarovať scénu“ a žiak v ňom ovláda čarodejníka Baltíka. Ten sa vie pohybovať v štyroch smeroch a ovládame ho tromi kartičkovými príkazmi – dopredu, otoč sa vpravo, otoč sa vľavo, a taktiež vie pred seba vyčarovať nejaký predmet (z ponuky). Čarodejník ihneď vykonáva zadané príkazy, preto je tento režim pre žiakov vhodný na pochopenie relatívneho pohybu a funkcie základných kartičiek.

Tretí režim je programátorský, volá sa „Programovať“ a delí sa dve časti – Začiatočník a Pokročilý. V režime „Programovať – Začiatočník“ (na Obr. 2) žiaci programujú čarodejníka. Program vytvárajú pomocou kartičiek. Okrem základných kartičiek z prvého

režimu pribudli aj kartičky, ktoré čarodejníka otočia presným smerom (sever, juh, východ, západ), možnosti opakovania (s presným počtom opakovaním alebo nekonečným opakovaním), vytvárania blokov (a ich následného opakovania) a ďalšie možnosti pre čarodejníka. Program sa spúšťa tlačidlom „spusti“ a v novom okne sa animuje pripravený program. Okrem celého programu sa dá spustiť aj označený blok príkazov.



Obr. 2 - Baltík, ukážka programovacieho módu

Posledný režim, „Programovať – Pokročilý“, ponúka možnosti bežné pre vyššie programovacie jazyky – premenné, čítanie vstupu z klávesnice, náhodné čísla, for cykly, podmienky a podobne. Už nie je potrebné programovať čarodejníka, v tomto režime sa dajú vytvárať aj programy bez neho, preto je vhodný pre starších žiakov a žiakov, ktorí sú v programovaní zdatnejší.

Baltík je pomerne využívaný na slovenských školách, keďže existuje aj v slovenskej jazykovej mutácii a zároveň vzniklo aj viacero učebníc, či metodických materiálov pre učiteľov.

### 1.4.2. EasyLogo

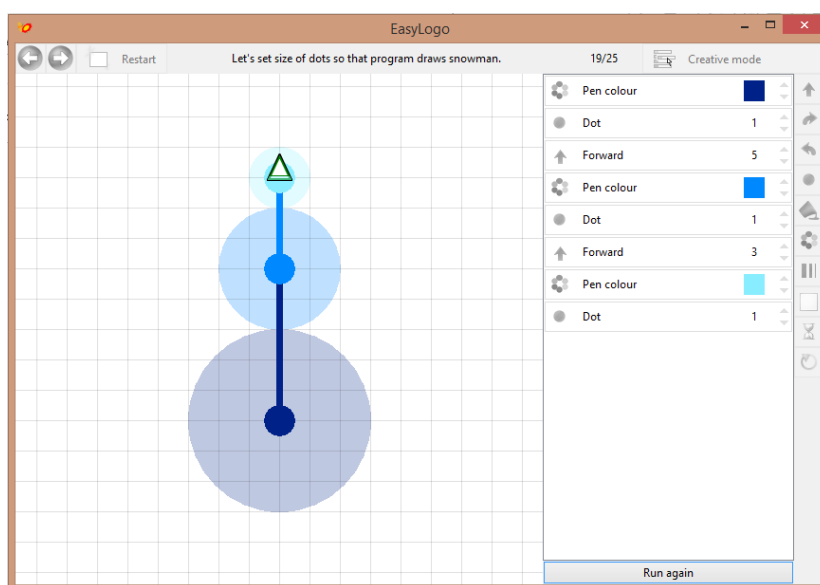
EasyLogo [10] (alebo v slovenskej jazykovej mutácii známe ako IzyLogo) je prostredie využívajúce princípy Loga, alebo Imaginu, v zjednodušenej verzii. Program obsahuje dva módy – jeden s predpripravenými aktivitami a druhý, takzvaný kreatívny.

V prvom móde je pre žiakov pripravených 25 aktivít. Prvých šesť oboznamuje žiakov s prostredím a s ovládaním postavičky, ktorá sa pohybuje v štvorčekovej ploche. Ovláda sa tromi príkazmi – dopredu, otoč sa vpravo a otoč sa vľavo. V tejto časti aktivít sa príkazy vykonávajú hneď po stlačení príslušnej ikony, takže žiak priamo vidí výsledok svojho programu.

Ďalšia časť aktivít sa už venuje programovaniu. Príkazy sa posúvajú do špeciálneho stĺpčeka. Pri každom z nich sa dá meniť parameter – nastaviť počet krokov dopredu, alebo upraviť uhol otočenia (vždy po 45°). Krok dopredu znamená posun na najbližší priesečník priamok na mriežke v smere natočenia, takže nie krok o presnú vzdialenosť. Rovnako ako v Imagine sa dá meniť farba a hrúbka pera, či maľovať body. Príkazy sa vykonávajú hneď po ich pridaní do programu, dajú sa vymazať, upravovať, či meniť ich poradie – vtedy sa ukáže iba výsledok, jednotlivé kroky sa nezobrazujú postupne. Existuje tu ale príkaz „čakaj“, vďaka ktorému ju možné pozorovať aj postupné vykresľovanie.

Niekoľko aktivít učí žiakov používať cykly, ďalšie obsahujú už predpripravený kód, v ktorom je ale chyba. Na obrazovke vidíme výstup programu a aj obrázok, ktorý by malo pero nakresliť, žiaci majú opraviť program, aby bol výstup správny. (Príklad je na Obr. 3.) V poslednej časti žiaci vytvárajú procedúry – úlohy na seba nadväzujú, ako napríklad vytvorenie domčeka, ktorému predchádza nakreslenie obdĺžnika a nakreslenie trojuholníka.

V kreatívnom móde môžu žiaci programovať voľne, je to vhodné hlavne na kreslenie rôznych obrázkov na štýl, hoci čiastočne oklieštenej, korytnačej grafiky.



Obr. 3 - EasyLogo, aktivita na opravu chýb



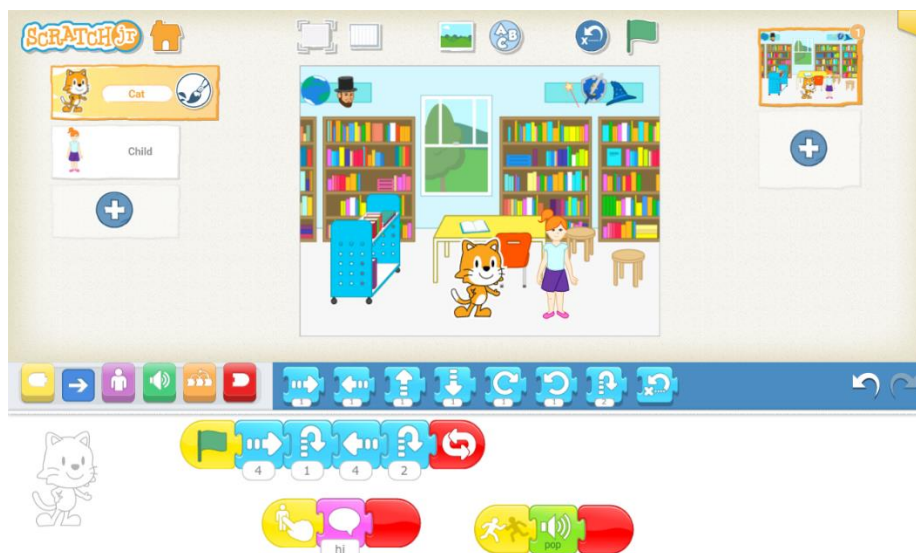
EasyLogo je vhodné pre žiakov, ktorí budú následne pokračovať s programovaním v Imagine. Učí relatívny pohyb, cykly a procedúry. A vďaka použitiu pripravených úloh aj voľného módu je pri zadávaní úloh možné využívať aj všetky stupne Bloomovej taxonómie.

### 1.4.3. ScratchJr

ScratchJr [11] je zjednodušenou verziou Scratchu určenou už tým najmenším. Je dostupný ako bezplatná aplikácia na tablety (Android aj iOS). Prostredie je podobné ako pri Scratchi – žiaci majú k dispozícii niekoľko pozadí, ktoré sa dajú priamo v aplikácii upravovať, taktiež viacero postavičiek a objektov, ktoré je možné programovať pomocou kartičiek pripomínajúcich puzzle. Pri programovaní je k dispozícii šesť skupín príkazov.

V prvom, označenom žltým tlačidlom, sú podmienky, kedy sa má postupnosť príkazov vykonať – pri spustení programu, pri stlačení postavičky, pri stretnutí s inou postavičkou, alebo pri prijatí správy. V modrej ponuke sa nachádzajú príkazy súvisiace s pohybom – jedná sa o absolútny pohyb, pri každom smere sa dá nastaviť počet krokov. Taktiež je tu aj možnosť natočenia postavičky – natočenie je ale vizuálne, nevlýva na ďalší smer pohybu. Postavička ďalej dokáže vyskočiť, či sa vrátiť domov – teda na svoju pôvodnú pozíciu. Tretia, fialová, ponuka sa venuje postavičke – vie prehovoriť ľubovoľné slovo, zmenšiť či zväčšiť sa, byť viditeľná alebo neviditeľná. Zelený súbor príkazov ponúka zvukové možnosti – okrem štandardného zvuku si žiaci môžu nahráť aj vlastné zvuky. V oranžovej ponuke nájdeme časovač, zastavenie postavičky, nastavenie rýchlosti, či opakovania. Červená ponuka obsahuje ukončenia príkazov – klasické (end) a nekonečné opakovanie.

Pre každú postavičku žiaci vedú vyskladať niekoľko typov udalostí v spodnej časti obrazovky, ako môžeme vidieť na Obr. 4. Pri kliknutí na zelenú vlajku v hornej časti aplikácie sa program spustí na celú obrazovku. V ScratchJr tak vedú žiaci vytvárať jednoduché multimediálne aplikácie, ktoré vedú navzájom zdieľať. Toto prostredie je vhodné pre školy, ktoré majú k dispozícii tablety a pre žiakov nižších ročníkov, ktorí neskôr budú programovať v Scratchi, keďže v tomto prostredí porozumejú základným príkazom a základnej funkcionalite a nebudú dezorientovaní príliš veľkým množstvom príkazov.



Obr. 4 - ScratchJr, ukážka projektu

## 1.5. Hry na rozvoj algoritmického myslenia

Okrem mini jazykov existuje aj viacero hier, ktoré u detí rozvíjajú algoritmické myslenie a učia ich základy programovania. Jedná sa o hry, v ktorých deti ovládajú postavičku pomocou príkazov – najjednoduchšie a najčastejšie sa vyskytujúce sa sú hry ovládané šípkami, zložitejšie, a z programovacieho hľadiska aj zaujímavejšie, sú hry v ktorých deti namiesto šípok používajú príkazy zobrazené pomocou ikon či obrázkov. Príkazy sa môžu vykonávať priamo, ako ich dieťa pridá do svojho „programu“, alebo až po spustení špeciálnym tlačidlom.

Vybrali sme si dve hry, ktoré môžu slúžiť žiakom na prvom stupni ako predpríprava k programovaniu v nejakom mini jazyku, alebo ho v prípade nedostatku času úplne nahradiť.

### 1.5.1. BotLogic

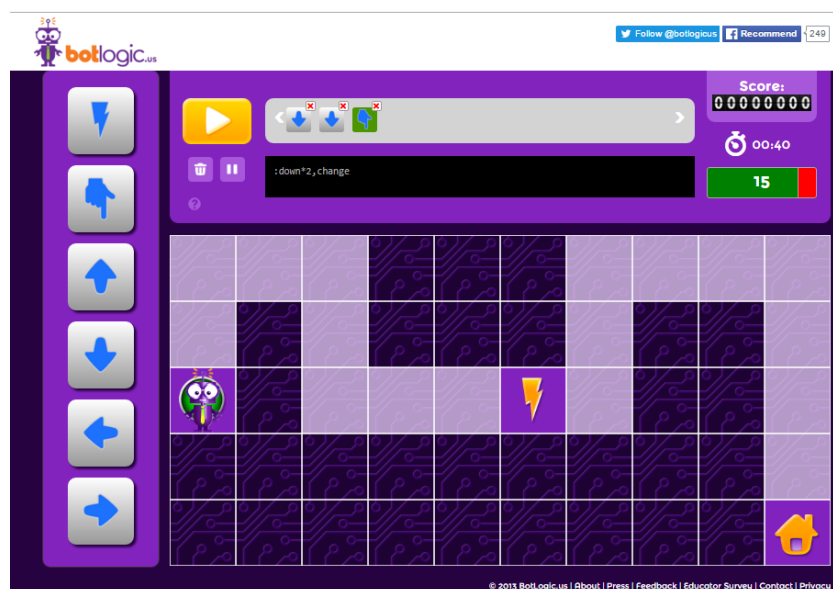
Hra BotLogic [12] je zasadená na jednoduchého dvojrozmerného labyrintu, v ktorom sa pohybuje robot a jeho cieľom je dostať sa do domčeka. Robota hráč ovláda vytvorením postupnosti príkazov – hore, dole, vpravo, vľavo, v ďalších úrovniach sa môžu pridať aj ďalšie príkazy, ako napríklad stlačiť. V hre je obmedzený počet príkazov, ktoré vie robot vykonať, vo vyšších úrovniach si ich však vie dobýjať zbieraním bateriek.

Zaujímavosťou je, že okrem vizuálnej postupnosti krokov, ktorá sa zobrazuje v hornej časti obrazovky, sa tesne pod ňou zobrazuje aj textová podoba príkazov. Príkazy tam hráči

môžu priamo dopisovať namiesto stláčania tlačidiel, ktoré pridávajú kartičky do postupnosti. Toto môže byť zaujímavé pre starších, alebo skúsenejších, žiakov, keďže syntax príkazov je jednoduchá a napríklad pre zopakovanie príkazu vpravo trikrát stačí napísať „right\*3“.

Program vie hráč postupne spúšťať a následne dopĺňať – tak vidí, kam sa až dostal robot a koľko krokov mu ešte zostáva, vid'. Obr. 5. Hra rozvíja logické myslenie – hráč musí nájsť najkratšiu cestu labyrintom, používa absolútny pohyb, takže sa hráč musí sústrediť hlavne na správny počet krokov a nie natočenie. V každej úrovni je časomiera, ktorá pôsobí ako motivátor.

Hra je dostupná na stránke <http://botlogic.us/play>, obsahuje dvadsať úrovní s postupne sa zvyšujúcou náročnosťou. BotLogic učí absolútny pohyb, pohyb po bludisku a pre zdatnejších žiakov môže byť dobrým štartom k textovej forme príkazov.



Obr. 5 - BotLogic, ukážka pokročilej úrovne

### 1.5.2. Lightbot

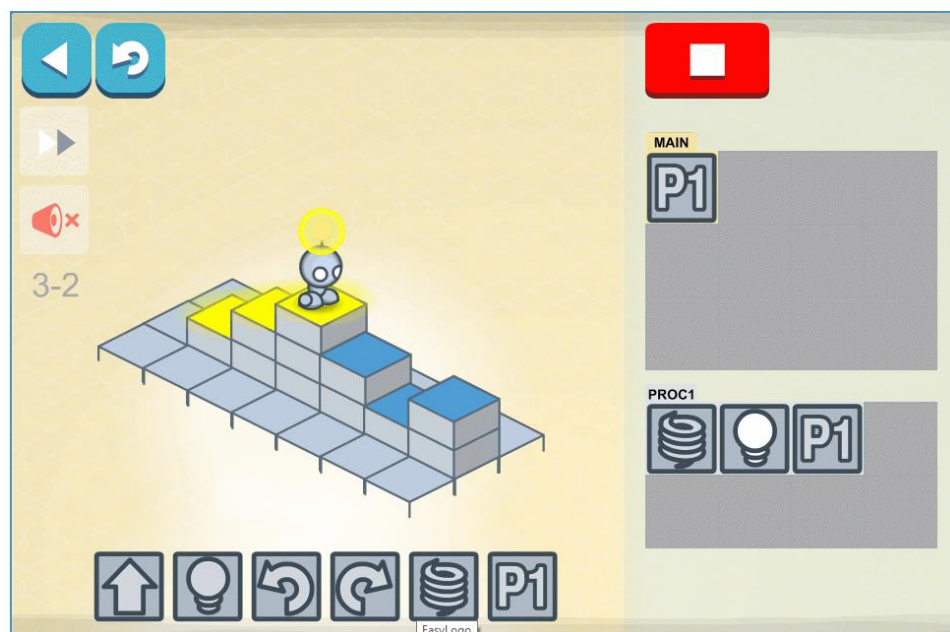
Lightbot [13] je jedna zo známejších hier, v ktorej hráč ovláda robota rozsvecujúceho žiarovky. Robot sa pohybuje v miestnosti po štvorcovanej ploche v štyroch smeroch. Na ploche sa môžu nachádzať prekážky a je tam minimálne jedno miesto, na ktorom musí rozsvietiť žiarovku. Úroveň je správne vyriešená, ak sú rozsvietené všetky žiarovky. Na ovládanie slúžia základné príkazy – dopredu, otoč sa vpravo, otoč sa vľavo, vyskoč a rozsviet' žiarovku. Všetky príkazy sú v grafickej podobe a hráč ich usporadúva na

predpripravené miesta. Po stlačení zeleného tlačidla „spusti“ robot začne postupnosť krokov vykonávať. Hráč ho vie kedykoľvek zastaviť, program zmeniť a znovu spustiť.

V hre sú tri skupiny úrovní – základy, procedúry a cykly. V základoch sa hráči postupne naučia, čo robia jednotlivé príkazy a vytvoria jednoduché programy. Pre každú úroveň je obmedzený počet príkazov, ktoré môžu hráči použiť, takže hra učí aj základy efektívnosti.

V úrovniach zo skupiny procedúry si hráč vie vytvoriť procedúru – postupnosť krokov – a tú následne zavolať v hlavnej procedúre. V náročnejších úrovniach prichádzajú ťažšie úlohy a aj možnosť použiť dve pomocné procedúry. Kvôli obmedzenému miestu na príkazy v každej procedúre (aj hlavnej) si musí hráč dopredu pripraviť stratégiu a procedúry naozaj použiť.

Posledná skupina úrovní – cykly, neučí cykly s pevným počtom opakovaním, či podmienené cykly, akoby mohol napovedať názov. Lightbot prináša do hry rekurziu. V úrovniach z tejto skupiny je za úlohu použiť príkaz P1, volajúci procedúru 1, priamo v jej tele. Hlavná procedúra potom obsahuje iba volanie procedúry P1, môžeme vidieť na Obr. 6 (vo vyšších úrovniach sú už pomocné procedúry dve). Jedná sa tak o zjednodušenú rekurziu – keďže neexistuje triviálny prípad, robot jednoducho skončí, keď sú všetky žiarovky rozsvietené.



Obr. 6 - Lightbot, príklad rekurzcie

Hra Lightbot je dostupná na webovej stránke <https://lightbot.com/hocflash.html>, pre viac úrovní je nutné kúpiť aplikáciu (aj pre mobilné zariadenia), voľne prístupné úrovne však pre žiakov prvého stupňa stačia, keďže niektoré sú už naozaj náročné. Hra je aj v slovenskej jazykovej mutácii.

Lightbot učí hravou formou základy programovania s využitím relatívneho pohybu, využíva aj procedúry a dokonca sa žiaci môžu stretnúť aj s rekurziou.,

## 2. Absolútny a relatívny pohyb

Ako sme si v prvej časti ukázali, niektoré prostredia využívajú na učenie základov programovania absolútny, iné relatívny pohyb. Z Piagetovej teórie kognitívneho vývoja vyplýva, že žiaci na prvom stupni základných škôl ešte nemajú dostatočne vyvinuté abstraktné myslenie a preto pre mnohých môže byť spočiatku relatívny pohyb mätúci. V počítačových hrách sa využíva skôr absolútny pohyb pomocou šípok a to sa prenáša aj do reálneho života.

V tejto časti sa pozrieme na úlohy v súťaži iBobor a ich výsledky a navrhujeme vlastnú sadu úloh na otestovanie náročnosti absolútneho a relatívneho pohybu, ktoré následne vyskúšame v praxi.

### 2.1. iBobor

iBobor [14] je medzinárodná súťaž, ktorej cieľom je pozdvihnúť záujem o informatiku medzi žiakmi základných a stredných škôl. Na Slovensku prebieha v piatich úrovniach, pre žiakov 3. a 4. ročníka základných škôl je určená kategória Bobríci.

Súťažiaci v tejto kategórii riešia 12 úloh – 4 ľahké, 4 stredné a 4 ťažké. Na ich vyriešenie majú 30 minút. Za správne zodpovedanú úlohu môžu získať 3, 6 alebo 9 bodov (podľa obťažnosti), za nesprávnu odpoveď sa im body odrátavajú – pre ľahké úlohy -1 bod, pre stredné -2 a pre ťažké -3. Za nezodpovedanú otázku sa im body ani nepričítajú, ani neodčítajú. Na začiatku všetci súťažiaci začínajú s 24 bodmi, takže minimum dosiahnutých bodov je 0, maximum je 96. Úspešným riešiteľom je každý, kto dosiahne aspoň polovicu kladných bodov, t.j. pre Bobríkov 60.

V úlohách žiaci riešia jednak otázky súvisiace s využívaním digitálnych technológií, s bezpečnosťou, či základnými informatickými zručnosťami, ale aj úlohy, ktoré sa zaoberajú algoritmickým myslením, programovaním a podobne. V niektorých úlohách sa od žiakov požaduje výber odpovede, na iné odpovedajú vpísaním textu a niektoré sú interaktívne.

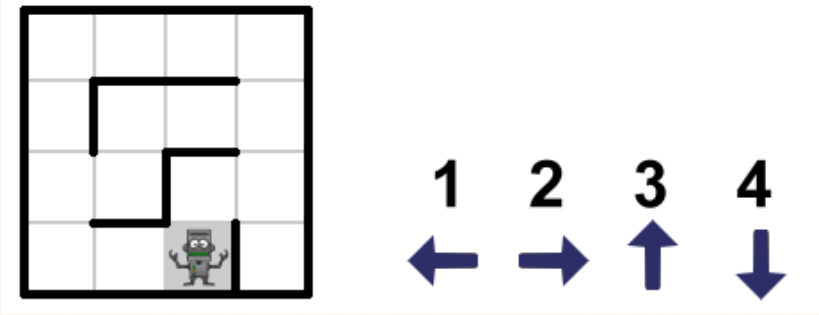
Z tejto kategórie sme z archívu vybrali niekoľko úloh zameraných na algoritmizáciu a programovanie.

#### 1.1.1. Labyrint – Bobríci 2013/14

V školskom roku 2013/14 v kategórii Bobrík súťažilo 12 448 súťažiacich, priemerný bodový zisk bol 42,64 bodu. Medzi úlohami sa vyskytla úloha Labyrint, na Obr. 7, ktorá

bola organizátormi označená ako ťažká. Úspešne ju vyriešilo približne 38% žiakov a približne 7% na ňu neodpovedalo.

**12. Labyrint**  
Tento robot sa programuje pomocou čísel 1, 2, 3 a 4. Prikaz 1 znamená, že sa presunie na susedné políčko vľavo. Prikaz 2 zasa, že sa presunie na susedné políčko vpravo, 3 hore a 4 dolu (pozri obrázok).



Ktorý z programov privedie robota späť na jeho štartovú pozíciu?

- 11332322441
- 32311413323
- 1132322441
- 3231141422

Obr. 7 - Úloha Labyrint, Bobríci 2013/14

V úlohe žiaci využívajú absolútny pohyb robota, ktorý je však zapísaný nie priamo pomocou šípok, ale cez čísla, čo úlohu sťažuje. Podľa Bloomovej taxonómie môžeme úlohu zaradiť k analyzovaniu – keďže žiaci musia jednak pochopiť príkazy, a následne analýzou možností prísť na to, ktorá je správna. Ďalším problémom, alebo chytákom, mohla byť odpoveď číslo tri – ktorá by robota dostala na jeho štartovnú pozíciu, ale iba ak by sa vpravo od neho nenachádzala stena medzi políčkami.

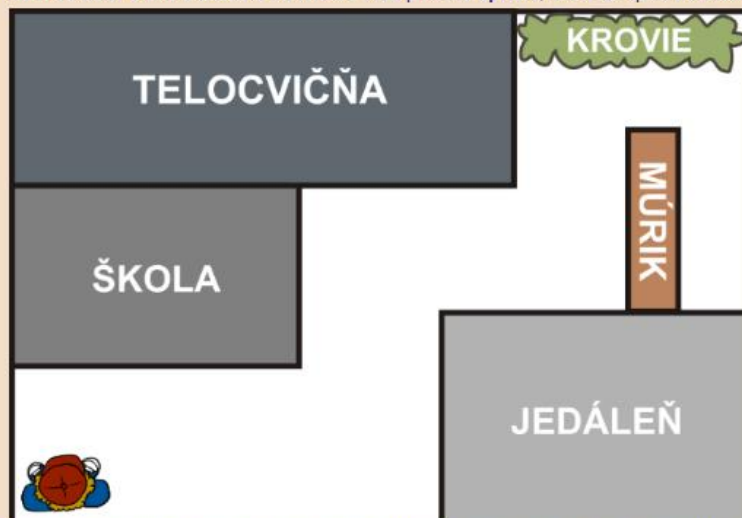
### 1.1.2. Riadenie robota – Bobríci 2013/14

Úloha Riadenie robota bola v súťaži v rovnakom roku. Hoci bola autormi označená ako stredná, správne ju vyriešilo ešte menej žiakov ako úlohu Labyrint – niečo okolo 32%. Na úlohu neodpovedalo asi 8% žiakov.

V úlohe sa vyskytuje relatívny pohyb a zároveň pohyb po prekážku (robot sa pohybuje, kým nenarazí), zadanie je na Obr. 8. Práve spojenie dvoch abstraktných pohybov mohlo viesť k väčšej náročnosti úlohy.

### 8. Riadenie robota

Deti sa hrajú s robotom. Keď povedia **dopredu**, robot kráča dopredu, až kým nenarazí. Keď povedia **vľavo**, robot sa otočí vľavo a zostane na mieste. Keď povedia **vpravo**, otočí sa vpravo a zostane na mieste.



Robot stojí v ľavom dolnom rohu dvora a je otočený ku škole. Deti chcú, aby prišiel ku krovu v pravom hornom rohu dvora.

Ktorá postupnosť povelov je správna?

- dopredu, vpravo, dopredu, vľavo, dopredu, vľavo, dopredu, vľavo, dopredu
- vpravo, dopredu, vľavo, dopredu, vľavo, dopredu
- vpravo, dopredu, vľavo, dopredu, vpravo, dopredu, vpravo
- dopredu, vpravo, dopredu, vľavo, dopredu, vpravo, dopredu, vľavo, dopredu

Obr. 8 - Riadenie robota, Bobríci 2013/14

### 1.1.3. Labyrint – Bobríci 2012/13

V kategórii Bobrík v roku 2012/13 bola medzi stredne ťažké úlohy zaradená aj úloha Labyrint. Túto úlohu správne vyriešilo iba 20% súťažiacich a približne 18% z nich neodpovedalo. Dokopy v tejto kategórii súťažilo 10 322 súťažiacich.

Zadanie bolo jednoduché, zostaviť kartičkový program, ktorý dovedol myš k cieľu (Vid'. Obr. 9). Úloha bola interaktívna – kartičky sa dali presúvať na políčka (okrem dvoch už vyplnených) pomocou drag and drop. Práve kvôli vyplneným kartičkám a faktu, že myška



spravila krok hore, vľavo a potom zas naspäť – vpravo, dole, mohla byť úloha pre žiakov veľmi ťažká, hoci sa zaoberala absolútnym pohybom.

**6. Labyrint**  
 Myška blúdila v labyrinte, až došla k syru.  
 Filip si značil jej cestu tak, že ukladal zľava doprava kartičky so šípkami podľa toho, ktorým smerom urobila krok na ďalšie políčko.  
 Kartičky mu popadali a len dve zostali na svojich miestach.  
 Pomôž mu doplniť cestu tak, aby šípky ukazovali, ako myška blúdila.

Obr. 9 - Labyrint, Bobríci 2012/13

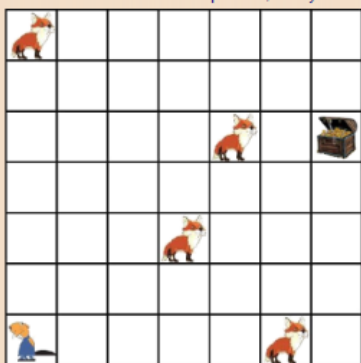
#### 1.1.4. Lov na poklad – Benjamíni 2012/13

Zaujímavé úlohy vieme nájsť aj vo vyššej kategórii – Benjamíni. Je určená žiakom piateho a šiesteho ročníka základných škôl a niekedy sa v nej vyskytnú rovnaké úlohy ako v kategórii Bobríci. Pre Benjamínov sú trochu iné podmienky ako pre Bobríkov – namiesto 12 úloh riešia Benjamíni 15 – 5 ľahkých za 3 body, 5 stredných za 4 body a 5 ťažkých za 5 bodov. Nesprávna odpoveď je pre ľahké úlohy hodnotená -1 bodom, pre stredné -1,33 bodu a pre ťažké -1,66 bodu. Počiatočný bodový zisk je 20 bodov, takže maximálne vedú súťažiaci získať 80 bodov a najmenej 0,05b.

V tejto kategórii bola v roku 2012/13 ako ťažká označená úloha Lov na poklad. Pre žiakov sa však ukázala ako ľahká, keďže ju zo 17 740 súťažiacich správne vyriešilo približne 70% a iba asi 2% na ňu neodpovedali. Zadanie je na Obr. 10.

### 11. Lov na poklad

Bobor Kubo chce získať poklad, ktorý strážia líšky. Na mape je zaznačené umiestnenie Kuba, líšok aj pokladu.



Líška zbadá Kuba, len ak stúpi na niektoré políčko tesne vedľa nej:



Akou cestou má ísť Kubo z jeho terajšej pozície, aby ho nevidela žiadna líška.

Skratky: D: dole, H: hore, P: vpravo, L: vľavo. Skratka: P3 znamená, že pôjde 3 políčka vpravo.

- P2H5P4D1
- H2P1H2P1H2P4D2
- P1H6P5D2
- P1H3P1H2P3D1P1

Obr. 10 - Lov na poklad, Benjamíni 2012/13

Úloha využíva absolútny pohyb zapísaný pomocou písmen a čísel. Súťažiaci si museli uvedomiť, že okolo každej líšky sa nachádza štvorec o rozmeroch 3 x 3 políčka, na ktoré bobor nesmie vojsť. Skúšaním postupností však hneď vedeli zistiť, kedy ich líška vidí a teda, ktoré sú zlé.

#### 1.1.5. Navigácia – Benjamíni 2012/13

Zaujímavou úlohou z hľadiska pohybu je úloha Navigácia z toho istého roku a kategórie, ktorá patrila medzi stredne ťažké. 41% zo žiakov ju vyriešilo správne, a približne 2% na ňu neodpovedali. Zadanie vidíme na Obr. 11.

Pohyb je síce absolútny, ale až v ôsmich možných smeroch znázornených pomocou kompasovej ružice. Okrem nájdenia cesty ku vlajke, museli súťažiaci uvažovať aj nad tým, ktorá z ciest je najkratšia, teda obsahuje najmenší počet krokov. Dve z postupností vedú loďku cez ostrov, zvyšné dve ju dovedú bez narazenia k vlajke. Žiaci museli vyhodnotiť aj to, že „najmenší počet krokov“ znamená, že musia zrátať čísla pri krokoch a nie počet textových príkazov.

**9. Navigácia**  
 Bobor Plavko je vo svojej loďke na jazere, na ktorom sú ostrovy. Chce sa ňou doplaviť až k vlajke. Jeho loďka má autopilot, ktorý ju dokáže riadiť tak, že ju podľa príkazov presúva z bodu do bodu v 8 rôznych smeroch.  
 Napr. príkaz 1 S znamená 1 krok smerom na sever, 2 SV znamenajú 2 kroky po uhlopriečke smerom na severovýchod.

Ktorá z nasledujúcich postupností príkazov pre autopilot určuje cestu s najmenším počtom krokov k vlajke bez toho, aby loďka narazila do niektorého ostrova?

2 SZ, 3 S, 3 Z  
 2 SZ, 2 Z, 1 SZ, 2 S  
 2 SZ, 2 Z, 1 S, 1 Z, 2 S  
 5 SZ

Obr. 11 - Navigácia, Benjamíni 2012/13

## 2.2. Testovanie motivácie

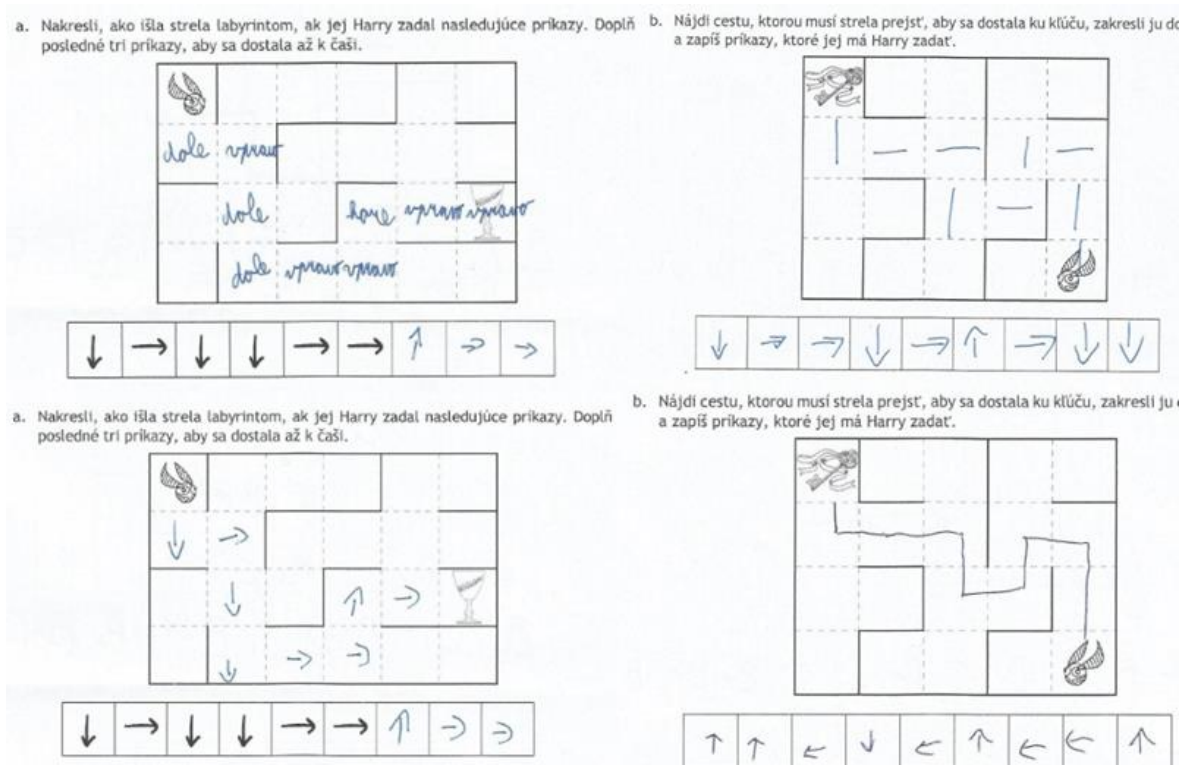
Okrem analýzy úloh zo súťaže iBobor sme spravili aj vlastnú malú analýzu. Pripravili sme pracovný list (v Prílohe 1), ktorý obsahoval úlohy zamerané na ovládanie postavičky v labyrinte. Dve úlohy boli zamerané na absolútny pohyb, dve na relatívny pohyb a v jednej úlohe si žiaci mohli zvoliť, akým typom pohybu budú danú úlohu riešiť. Prvá z dvojíc úloh obsahovala už niekoľko inštrukcií pre ovládanú postavičku (pre absolútny pohyb začarovaná lietajúca zlatá strela, pre relatívny pohyb kocúr) a žiaci mali na pripravené miesto doplniť tri príkazy. Z hľadiska Bloomovej taxonómie tak žiaci museli pochopiť, ako jednotlivé príkazy fungujú a následne tieto poznatky aplikovať, aby úlohu doriešili. Zároveň vyplnená časť slúžila na to, aby sa žiaci oboznámili so spôsobom riešenia pracovného listu. V druhej úlohe už museli nájsť cestu v bludisku a celú ju zapísať pomocou príkazov.

Testovania sa zúčastnilo 17 žiakov piatych ročníkov z rôznych škôl z Bratislavy a okolia, ktorí sa vo vzdelávacom centre pripravovali na prijímacie pohovory na osemročné gymnáziá. Pracovný list (viď. príloha 1) dostali v papierovej podobe a riešili ho samostatne. Nižšie uvádzame stručné výsledky. Celý opis k jednotlivým úlohám sa nachádza v [15].

Úlohy s absolútnym pohybom väčšine žiakov nerobili problémy. Prvú úlohu správne vyriešilo 16 žiakov, druhú malo celú správne 11 žiakov. Pri druhej úlohe sa 4 žiaci pomýlili

pravdepodobne rovnakou pozíciou začiatku bludiska v prvej úlohe a koncom bludiska v druhej, preto túto úlohu riešili od konca na začiatok bludiska. Inak bolo ich riešenie správne. Zvyšní dvaja žiaci mali v riešení iba jednu chybu.

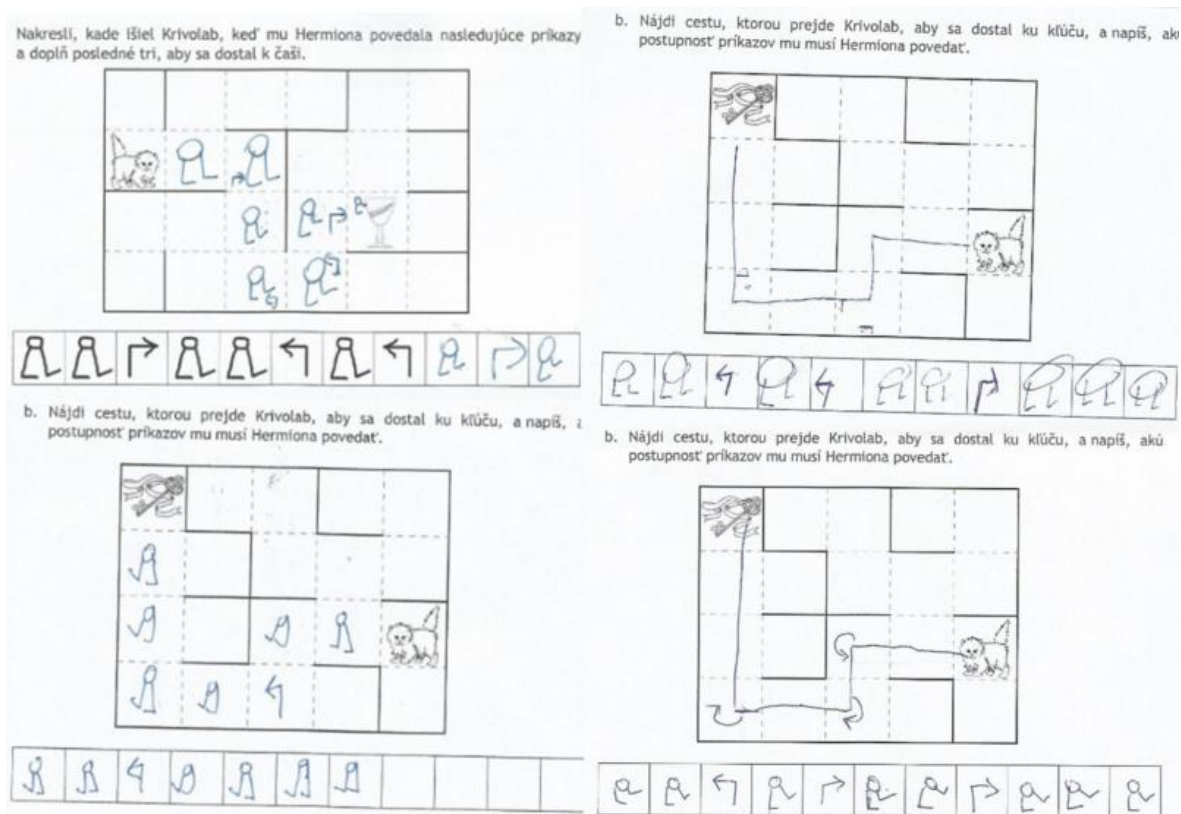
Zaujímavé boli rôzne spôsoby, ako si žiaci zakresľovali cestu postavičky v labyrinte. Jeden žiak písal priamo do štvorčekov v bludisku textové príkazy, dvaja (v prvej úlohe traja) žiaci cestu nezakreslili. Zo zvyšných 14 žiakov jedna polovica cestu zakreslila pomocou lomených čiar a druhá polovica pomocou šípok. Niektoré z riešení sú zobrazené na Obr. 12.



Obr. 12 - Ukážky žiackych riešení pre úlohy s absolútnym pohybom

Výsledky úloh s relatívnym pohybom dopadli horšie. Prvú úlohu vyriešili správne desiaty žiaci, jeden žiak svoje riešenie nedokončil, dve riešenia boli úplne nesprávne a zvyšní štyria žiaci túto úlohu neriešili. Pri tejto úlohe sme našli súvis výsledkov so zakresľovaním cesty. Štyria žiaci, ktorí cestu nezakreslili, mali úlohu buď nesprávne, alebo neriešenú. Piaty žiak cestu zakreslil pomocou príkazov – z toho si štyria, ktorí mali aj správne riešenia, uvedomili, že príkaz otočenia sa musí nachádzať na rovnakom políčku ako krok dopredu. Jeden žiak ale každý príkaz zaznačoval do samostatného políčka v bludisku a nakoniec úlohu nedokončil. (Vid' Obr. 13.)

Druhú úlohu vyriešilo správne osem žiakov, jeden žiak mal opäť opačné riešenie (od cieľa k štartu), dvaja žiaci spravili malé chyby (jedno zlé otočenie) a šesť riešení bolo nesprávnych. Počty pre jednotlivé zakresľovania boli približne rovnaké ako v predchádzajúcej úlohe, hoci žiaci medzi týmito dvoma úlohami často menili spôsob zakresľovania cesty. Opäť sa ale potvrdilo, že žiaci, ktorí mali zlé riešenia, nemali zakreslenú cestu alebo do bludiska písali zle príkazy.



Obr. 13 - Ukážka žiackych riešení pre úlohy s relatívnym pohybom

V poslednej úlohe si žiaci mohli zvoliť, ktorú z postavičiek (a teda aj typov pohybu) použijú. Všetci si vybrali absolútny pohyb, z toho iba jedno riešenie bolo nesprávne, dve riešenia obsahovali malé chyby a jeden žiak aj túto úlohu riešil opačne. Zvyšní trinásť žiaci mali správne riešenie. Žiačka, ktorá mala nesprávne riešenie, zakresľovala cestu pomocou šípok, ale v bludisku ich mala posunuté.



Graf 1 - Vyhodnotenie pracovného listu

Celkové výsledky znázorňuje Graf 1. Štyria žiaci vyriešili správne všetky úlohy, piati žiaci mali v nejakej z úloh drobné chyby (napr. ju riešili opačne, alebo mali zle jedno otočenie), piati žiaci mali jednu alebo dve úlohy nesprávne vyriešené a traja žiaci mali nesprávne vyriešené viac ako dve úlohy. Každý zo žiakov mal správne vyriešenú aspoň jednu úlohu. Testovania sa zúčastnilo 6 chlapcov – z toho po jednom dve úlohy nesprávne vyriešené, jedna úloha nesprávne vyriešená a celý pracovný list bez chyby. Zvyšní traja mali v riešení iba drobné chyby. Z toho vyplýva, že viac ako jednu chybnú úlohu malo päť dievčat.

Z výsledkov nášho prieskumu vyplynulo, že labyrinty sú vhodnou motiváciou na absolútny pohyb, ale na použitie s relatívnym pohybom postavičky sa nehodia. Predpokladáme, že problémy s otočením pri relatívnom pohybe, odstráni okamžité vyhodnocovanie, pri ktorom budú žiaci vidieť, kde sa postavička nachádza a akým smerom je otočená.

### 2.3. GEVIN

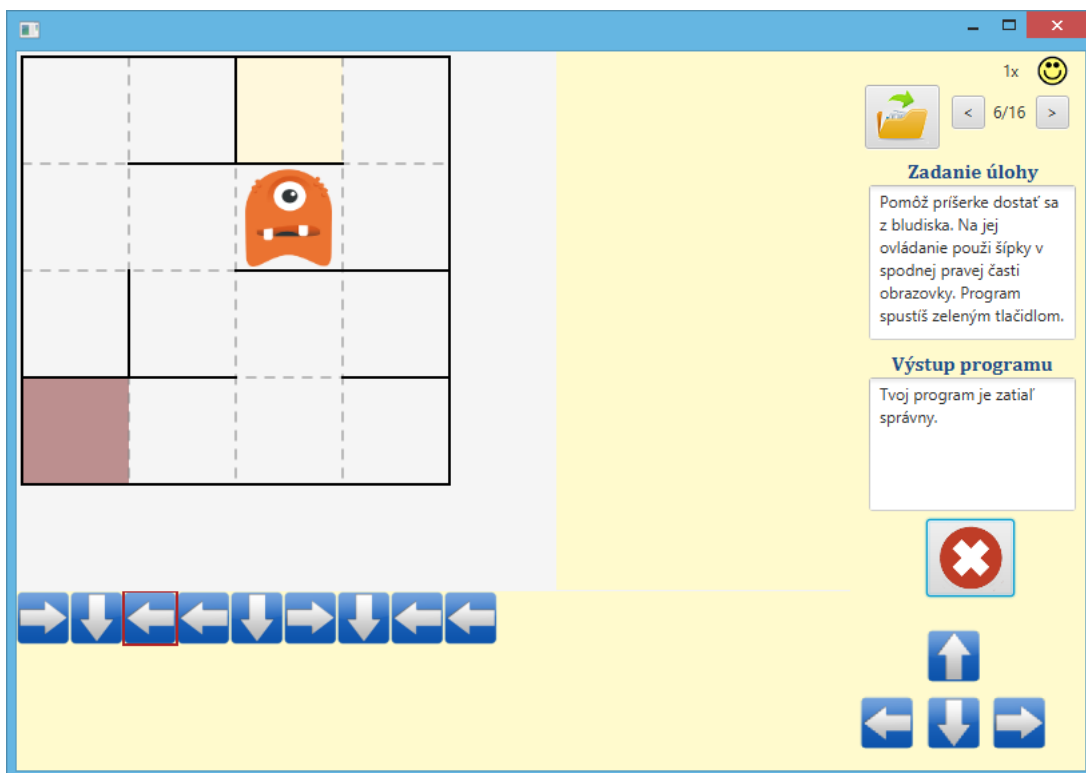
GEVIN (Generátor úloh na vyučovanie informatiky) je edukačné prostredie určené žiakom prvého stupňa základných škôl a ich učiteľom, ktoré vzniklo v rámci našej bakalárskej práce [16]. Ponúka dve časti aplikácie – jedna je určená pre učiteľov, v ktorej môžu vytvárať úlohy pre žiakov, a druhá časť je určená žiakom, v ktorom môžu riešiť učiteľmi vytvorené úlohy. V softvéri sa nachádza trojica šablón, do ktorých učiteľ iba zadá vstupné parametre a softvér vygeneruje úlohu. Prvá šablóna generuje pohyb postavičky v labyrinte, druhá šablóna je zameraná na zbieranie predmetov v štvorčekovej sieti a tretia

šablóna umožňuje kreslenie v štvorčekovej sieti pomocou 8-smerného pohybu (otáčanie po 45°). Tieto úlohy môže učiteľ ďalej upravovať a nastavovať im ďalšie vlastnosti, ako je napríklad ovládanie programu (priamy režim pomocou šípok a dva programátorské režimy – absolútny kartičkový pohyb a relatívny kartičkový pohyb), vyhodnocovanie programu (okamžité alebo vyhodnocovanie celého programu), ale aj hľadanie najkratšej cesty, či zbieranie špeciálnych predmetov. Samozrejmosťou pri každej úlohe je nastavenie zadania a obrázku postavičky, pri zbieraní predmetov aj výber obrázkov pre predmety.

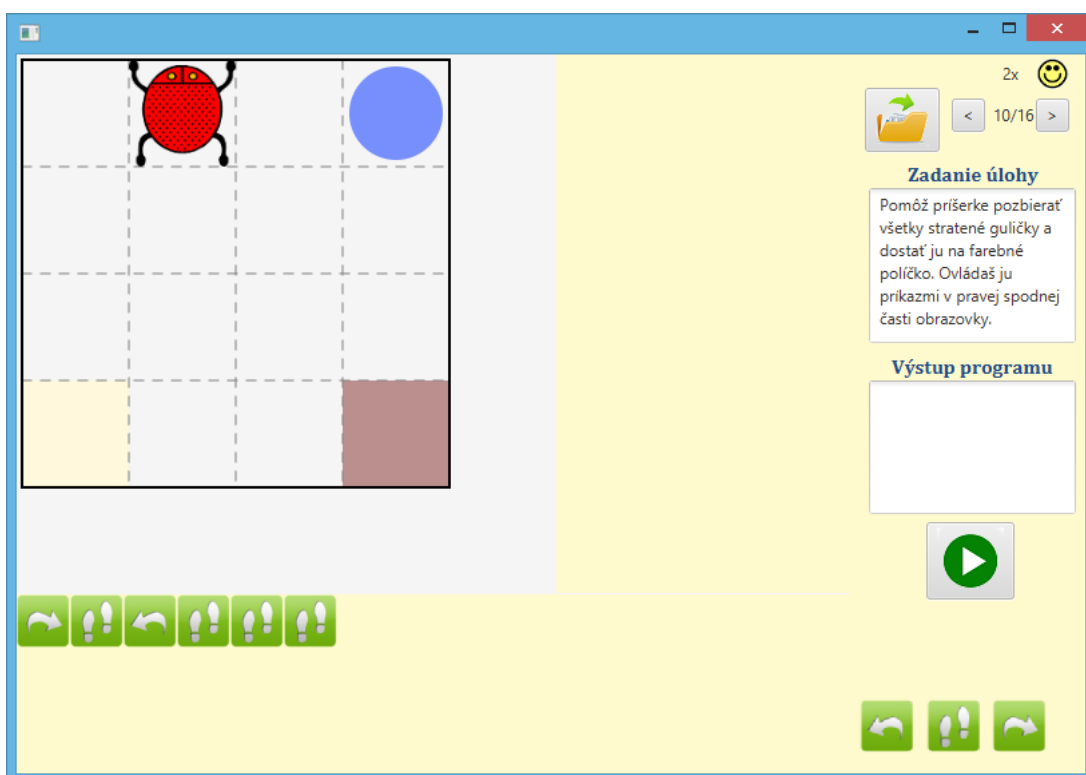
Softvér GEVIN sme pre účely tejto práce upravili. Jednotlivé režimy sme oddelili a žiacky režim prešiel väčšími zmenami. Okrem zmeny kartičiek pre jednotlivé programátorské režimy sme úplne zmenili vizuálnu stránku používateľského rozhrania. Žiacka časť aplikácie bola preprogramovaná v JavaFX, čím sa podarilo zmodernizovať jej vzhľad. Aplikácia však stále dokáže spolupracovať so starou verziou a učiteľskou časťou softvéru, ktorú sme nemeni.

V aktuálnej verzii žiacka časť prostredia ponúka načítavanie dvoch typov úloh – labyrintov a zbierania predmetov. Podporuje všetky tri typy ovládania a obe vyhodnocovania. Kartičky na ovládanie postavičky obsahujú iba základné príkazy na pohyb (hore, dole, vpravo, vľavo, resp. krok dopredu, otoč sa vpravo, otoč sa vľavo) a nie cykly, či špeciálne kartičky na zbieranie predmetov.

Softvér umožňuje žiakom spúšťať aj sady úloh. Medzi úlohami sa môžu jednoducho presúvať, vidia poradové číslo úlohy aj počet správne vyriešených úloh. S tým súvisí ďalšia vlastnosť prostredia GEVIN – kontroluje žiacke riešenie, nedovoľuje im napríklad posielat' postavičku cez steny. V prípade chyby a aj správneho riešenia sa žiakovi zobrazí oznámenie. Pokiaľ žiak spúšťa čiastočný program, jeho výsledok vidí v dvoch podobách – postavička animuje postupne svoj pohyb a popritom sa zvyrazňuje práve vykonávaná kartička, pokiaľ je program správny alebo sa v ňom nachádza chyba, textová informácia sa zobrazí v textovom poli „Výstup programu“.



Obr. 14 - Ukážka prostredia GEVIN  
 spustená je šiesta úloha zo 16, pričom žiak už správne vyriešil jednu úlohu; aktuálne sa vykonáva program s absolútnym pohybom, posledná kartička, ktorá sa vykonala je zvýraznená



Obr. 15 - Ukážka prostredia GEVIN - úloha s relatívnym pohybom a okamžitým vyhodnocovaním



### 3. Testovanie

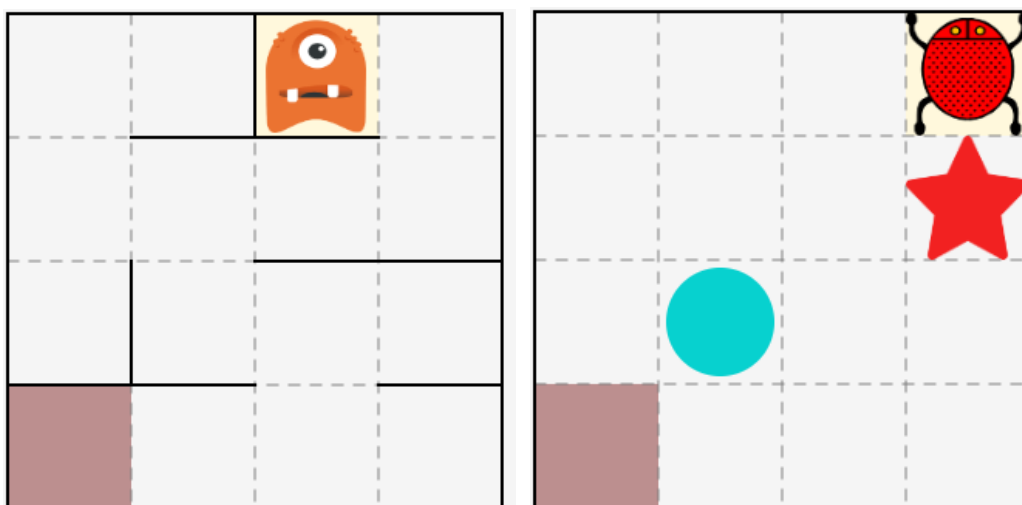
V tejto kapitole sa budeme venovať postupnosti úloh, ktoré sme navrhli na testovanie so žiakmi prvého stupňa. Následne opíšeme priebeh testovania a jeho výsledky, ktoré zhodnotíme a zanalyzujeme.

#### 3.1. Návrh úloh

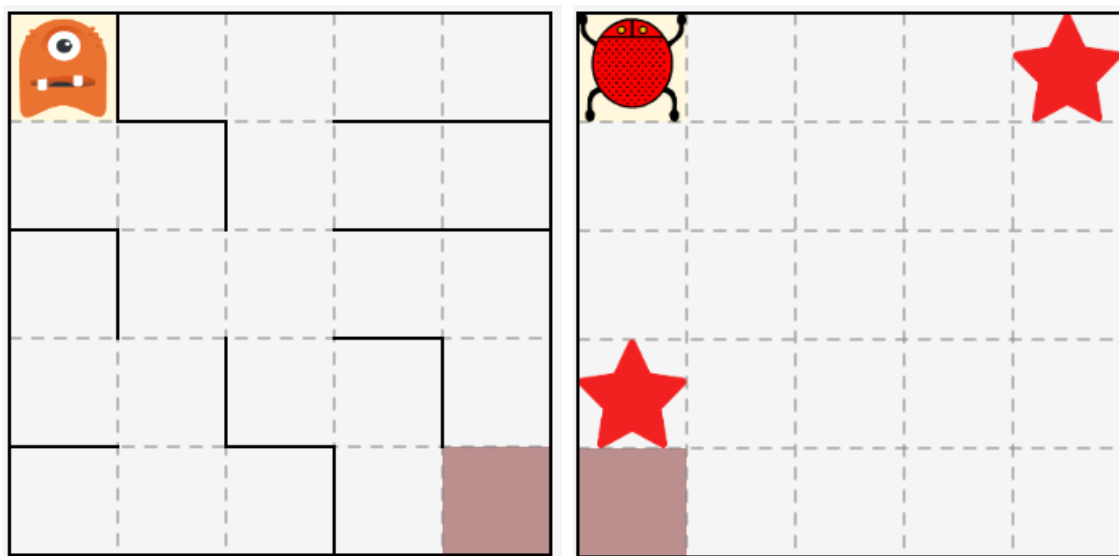
Na testovanie sme pripravili 16 úloh – 8 využívalo na riadenie postavičky absolútny pohyb a 8 relatívny pohyb. V úlohách s absolútnym pohybom riadili žiaci príšerku v bludisku. Prvé dve úlohy boli v priamom režime – príšerku ovládali šípkami na klávesnici. V ďalších úlohách používali kartičky na absolútny pohyb. Pri tretej a štvrtej úlohe sa príkazy vyhodnocovali okamžite, v zvyšných štyroch úlohách mali žiaci pripraviť celý program a ten následne spustiť. Niektoré z labyrintov sa zopakovali, t. j. žiaci riadili postavičku v rovnakom labyrinte najprv s okamžitým vyhodnocovaním (alebo v priamom režime) a následne aj pri vyhodnocovaní celého programu.

Druhá polovica úloh využívala relatívny pohyb. Štyri úlohy boli opäť s okamžitým vyhodnocovaním a štyri s vyhodnocovaním celého programu. Pri relatívnom pohybe žiaci zbierali v štvorcovej ploche predmety, ktoré mala následne postavička „odnieť“ na určené políčko. Aj v tejto skupine úloh sa niektoré zopakovali pre obe vyhodnocovania.

V oboch skupinách boli štyri úlohy (dve s okamžitým vyhodnocovaním a dve s vyhodnocovaním po spustení) v menšej ploche 4x4 (Obr. 16), zvyšné úlohy boli na ploche 5x5 (Obr. 17).



Obr. 16 - Ukážky úloh na ploche veľkosti 4x4 (vľavo labyrint s absolútnym pohybom, vpravo zbieranie predmetov s relatívnym pohybom)



Obr. 17 - Ukážky úloh vo veľkosti 5x5 (vľavo labyrint s absolútnym pohybom, vpravo zbieranie predmetov s relatívnym pohybom)

Žiakom sa po vyriešení úlohy zobrazilo oznámenie, že ju vyriešili správne, a v hornej časti okna sa im pridal počet smajlíkov. V prípade, že mali v programe zlú kartičku a viedli postavičku cez steny, zobrazilo sa im upozornenie a program im nedovolil (pri okamžitom vyhodnocovaní) pridať ďalšiu kartičku, kým svoju chybu neopravia. Pri vyhodnocovaní po stlačení tlačidla „spustiť“, sa postavička začala postupne animovať s tým, že sa zvýrazňovala práve vykonávajúca sa kartička. Ak kartička viedla postavičku cez stenu, program zastal, chybná kartička zostala zvýraznená a objavila sa chybová hláška. Po spustení programu, keď žiaci zistili, že postavička sa hýbe inak, ako chceli, mohli vykonávanie programu zastaviť, upraviť ho a potom opäť spustiť.

Medzi úlohami prepínali pomocou šípok v hornej časti okna. Rovno videli aj číslo úlohy, ktorú aktuálne riešia. Pri návrate na predchádzajúcu úlohu sa im nezobrazoval ich program, ale mohli úlohu riešiť odznova. Takýto postup sme zvolili preto, aby žiaci museli rozmýšľať nad riešením a nie „odpisovať“ kartičky, ktoré už raz zadali. Treba poznamenať, že druhýkrát správne vyriešená úloha im nepridala počet smajlíkov.

Pre každého žiaka a každú z úloh sa do súboru ukladal názov úlohy a číselná reprezentácia riešení. Pri okamžitom vyhodnocovaní to bola číselná reprezentácia každého príkazu, ktorý zadali, pri vyhodnocovaní programu po spustení sa do súboru zapísala každá postupnosť príkazov, ktorú žiak spustil. Vďaka tomu sme mohli zistiť, akým štýlom riešili dané úlohy, a aj to, ako pracovali s chybami.

### 3.2. Priebeh a výsledky testovania

Testovania sa zúčastnilo 17 žiakov 3.B triedy na Základnej škole Jozefa Horáka v Banskej Štiavnici, z čoho bolo 9 chlapcov a 8 dievčat. Tretíakov sme zvolili preto, že žiaci štvrtého ročníka na tejto škole už niekoľko mesiacov programujú v Baltíkovi a EasyLogu a teda zisťovanie, ako budú riešiť úlohy so zameraním na pohyb postavičky, by bolo ovplyvnené ich skúsenosťami. Žiaci 3.B doteraz na hodinách informatickej výchovy neprogramovali, riešili však už úlohy, v ktorých ovládajú postavičku v labyrinte pomocou klávesnice (jednalo sa o úlohy vo vzdelávacom programe GCompris [17]).

Žiaci riešili úlohy samostatne, pokiaľ niečomu nerozumeli, alebo mali pocit, že program nerobí to, čo má, ozvali sa. Väčšina problémov však spočívala iba v tom, že si neprečítali zadanie a očakávali, že budú riešiť úlohu rovnako, ako tú predchádzajúcu. V programe sa zorientovali celkom rýchlo, ovládanie im nerobilo problémy. Špecifické veci ako presúvanie a odstraňovanie kartičiek, sme im s ich pani učiteľkou vysvetlili, keď sa na to pýtali. Väčšina žiakov ale radšej riešila úlohu odznova, ak si vyžadovala viac zásahov, podľa ich vyjadrení sa im ľahšie vytváral nový program, akoby mali opravovať už existujúci.

Zaujímavosťou bolo, že rýchlosť a správnosť riešenia v tejto skupine súvisela aj s pohlavím. Chlapci vyriešili sadu úloh oveľa skôr, žiadna z nich im nerobila väčšie problémy. Dievčatá riešili úlohy dlhšie, pri posledných strávili veľa času a niektoré z nich potrebovali pomoc pri riešení. Spolu s ich pani učiteľkou sme sa ich snažili naviesť na to, aby si našli chyby v programe, alebo vyskúšali spustiť jednotlivé kroky, pri ktorých si neboli isté.

Z výsledkov mini ankety po vyriešení sady úloh, v ktorej mali žiaci pomocou smajlíka znázorniť, ako ťažké sa im zdali úlohy a ako veľmi rozumeli, čo majú v programe robiť, chlapci odpovedali, že úlohy boli ľahké. Z rozhovorov s nimi vyplynulo, že úlohy sa im zdali veľmi ľahké, trochu ťažšie im prišli posledné úlohy, ale radi by riešili ešte viac a ťažších úloh. S relatívnym pohybom problémy nemali, čo mohlo byť spôsobené tým, že sa hrajú počítačové hry, v ktorých sa často relatívny pohyb vyskytuje.

Dievčatá v ankete uvádzali, že úlohy boli ľahké, dve uviedli, že boli stredne ťažké, a jedna žiačka, ktorej riešenie úloh trvalo najdlhšie, uviedla, že úlohy boli ťažké. Dievčatá, ktoré hodnotili úlohy ako stredne ťažké, rovnako hodnotili aj ovládanie programu a pochopenia inštrukcií. Podobne hodnotili ovládanie programu aj dvaja chlapci, ktorí namiesto vymazávania kartičky presúvali, pričom si neuvedomili, že sú stále súčasťou programu,

a teda následne sa nevedeli zbaviť upozornení, že postavička nevie chodiť cez steny. Úlohy ale vyriešili od začiatku ešte raz, tentoraz bez problémov.

Zvyšní žiaci hodnotili ovládanie a zrozumiteľnosť úloh ako dobré, program sa im podľa ich slov páčil a radi by v ňom robili aj ďalšie úlohy.

### 3.2.1. Analýza žiackych riešení – absolútny pohyb

Prvé dve úlohy, v ktorých žiaci ovládali postavičku pomocou šípok na klávesnici, vyriešili veľmi rýchlo. Pár žiakov sme museli upozorniť na to, že hrubšie čiary znamenajú stenu, cez ktorú postavička chodiť nevie. Nasledujúce dve úlohy zvládli s podobnou ľahkosťou, jediný problém nastal pri zmene ovládania, keď žiaci chceli automaticky riešiť úlohu pomocou šípok, čo nefungovalo a tak mali pocit, že program je zlý.

Pri úlohách, v ktorých žiaci ovládali postavičku pomocou absolútneho pohybu s vyhodnocovaním po spustení tlačidla, si väčšina žiakov pripravila celý program, ktorý následne spúšťali a iba v prípade, že im nejaký krok chýbal, alebo mali v programe chybu, ho neskôr upravovali. Pri prvom takomto príklade (celkovo piaty v sade) sme zaznamenali zvýšený výskyt postupného spúšťania programu po pridaní jednej alebo dvoch kartičiek, ďalšie tri príklady s rovnakým ovládaním však žiaci riešili v priemere na dve spustenia. Výnimkou sú tri dievčatá. Každá z nich riešila všetky štyri príklady s absolútnym pohybom (príklady 5 až 8) tak, že po pridaní akejkoľvek kartičky program spustili.

<pre>absolCP3.lab 0 01 010 0100 01001 010011</pre>	<pre>absolC3.lab 1 10 101 1010 10100 101000 1010001 10100011</pre>	<pre>absolCP4.lab 0 01 012 0122 01221 012210 0122101 01221012 012210122</pre>	<pre>absolC4.lab 0 00 001 0012 00121 001211 0012111 00121110 001211103 0012111030 001211103010</pre>
--	--	---	--

Obr. 18 - Ukážka riešenia úloh 5 až 8 (postupne), (0 - krok vpravo, 1 - krok dole, 2 - krok vľavo, 3 - krok hore)

Na Obr. 18 vidíme textovú reprezentáciu spúšťaných programov jednej z trojice dievčat, kde 0 značí krok vpravo, 1 krok dole, 2 krok vľavo a 3 krok hore. Jeden riadok znamená

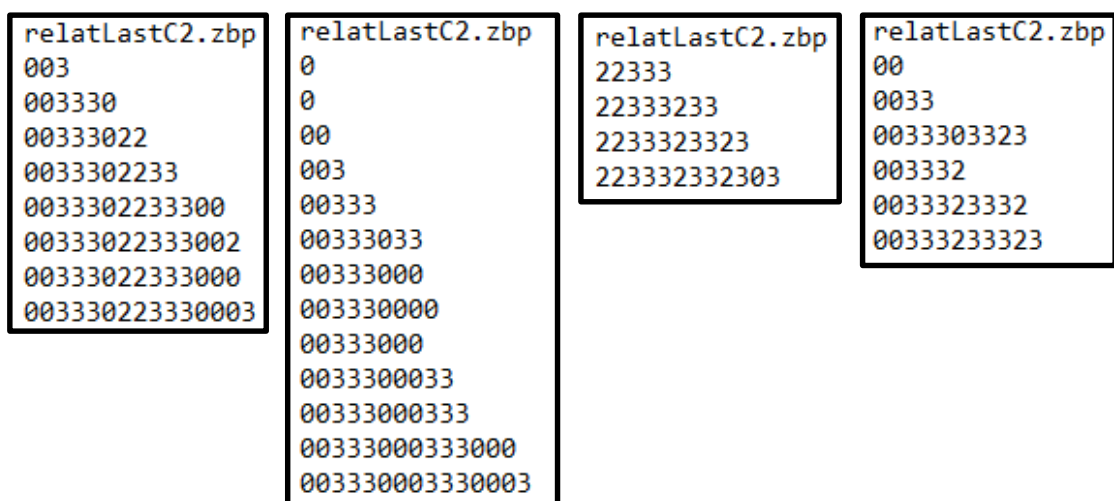
jedno spustenie programu. Všetky tieto žiaci rovnako riešili aj úlohy s relatívnym pohybom a vyhodnocovaní po stlačení tlačidla.

### 3.2.2. Analýza žiackych riešení - relatívny pohyb

Pri relatívnom pohybe riešila väčšina žiakov príklady na viac spustení ako pri absolútnom pohybe. Opäť, pri prvej takejto úlohe (úloha 12), bol výskyt spustení väčší, ako pri ostatných. Všimli sme si, že väčšinou program spúšťali po pridaní kartičky s otočením. Často sa stávalo, že namiesto otočenia vpravo použili radšej trikrát otočenie vľavo (a naopak). Súdiac z ich riešení, toto bolo spôsobené tým, že pri vyhodnotení sa postavička otočila inak, ako očakávali a namiesto vymazania danej kartičky a pridania správnej radšej pridali ďalšie dve otočenia.

Na Obr. 19 vidíme niekoľko korektných riešení úlohy 14. V prvom riešení vidíme, že daná žiacka používala kombináciu otočenie vpravo a následne otočenie vľavo, čím vlastne korigovala svoje riešenie. V druhom riešení na rovnakom obrázku zas žiacka ani raz nepoužila otočenie vľavo. (Takýchto žiakov bolo viac.) Namiesto neho používala trikrát otočenie vpravo, čo ale nebolo spôsobené tým, že nevedela chybu opraviť inak – keďže iné chyby dokázala odstrániť vymazaním kartičiek.

Ďalšie dve riešenia sú chlapčenské. V treťom riešení daný žiak používal oba typy otočenia a vidíme, že program spúšťal vždy po pridaní krokov dopredu, aby následne vedel určiť otočenie a ďalšie kroky. Z posledného riešenia na Obr. 19 vidíme, že daný žiak si dané zlé otočenie opravil, vyskúšal a následne dorobil zvyšok programu.



Obr. 19 - Ukážky riešení úlohy s relatívnym pohybom (0 – otočenie vpravo, 2 – otočenie vľavo, 3 – krok dopredu)

### 3.3. Návrhy na zlepšenie

Pri testovaní so žiakmi a pri následnej analýze ich riešení sme prišli na viacero problémov, ktoré u niektorých žiakov nastali. Ako sme už spomínali vyššie, niektorí žiaci spúšťali program po každom kroku alebo po veľmi malých skupinách príkazov. Toto je vhodné pri učení sa relatívneho pohybu a vytvárania programov, neskôr by už mali žiaci zvládnuť vytvoriť celý program naraz a následne iba opravovať prípadné chyby. Z tohto dôvodu by možno bolo dobré doplniť do programu povolený počet spustení pre každú jednu úlohu, čo by žiakov motivovalo k väčšiemu premýšľaniu pri vytváraní postupnosti krokov pre postavičku.

Rovnako sme si všimli, že niektorí žiaci nepoužívali obe otočenia, prípadne často otočili postavičku trikrát jedným smerom namiesto toho, aby ju otočili raz opačne. Toto by sa taktiež dalo odstrániť pomocou motivujúceho a herného obmedzenia – stanoviť maximálny počet kartičiek, ktoré môžu v danej úlohe použiť. Takto by sme žiakov prinútili plánovať a svoje chyby opravovať pomocou odstránenia alebo presunutia kartičiek a nie tak, že programovali postavičke pohyb späť.

Viacero žiakov spomínalo, že by uvítali viac úloh, ktoré by mohli byť zložitejšie. Na použitie vo vyučovaní by preto bolo vhodné pripraviť viac úloh, pričom spolu s vyššie uvedenými obmedzeniami na počet kartičiek a počet spustení programu, by bolo vhodné, aby sa žiaci na ďalšiu úlohu dostali až po úspešnom zvládnutí predchádzajúcich.

Ak by sa program používal v bežnom vyučovaní bola by vhodná úvodná diskusia žiakov s učiteľom, v ktorej by sa rozprávali o rôznych typoch pohybu, vysvetlili si, prečo je vhodné používať čo najmenej krokov a ako opravovať chyby. Podľa výsledkov testovania, pri ktorom žiadna podobná diskusia so žiakmi nebola a žiaci tieto úlohy riešili intuitívne, si myslíme, že podobné úlohy v prostredí GEVIN by mohli slúžiť ako dobrá propedeutika k programovaniu v nejakom z detských programovacích prostredí.

## Záver

V našej práci sme sa venovali absolútnemu a relatívnemu pohybu vo vyučovaní programovania na prvom stupni základných škôl. Vychádzali sme jednak zo štátnych vzdelávacích programov, ale aj z Piagetovej teórie kognitívneho vývinu, podľa ktorej žiaci prvého stupňa patria do štádia konkrétnych operácií, v ktorom ešte nie je plne rozvinuté abstraktné myslenie. Pri programovaní je však potrebné, aby si žiaci vedeli predstaviť, ako má program fungovať, preto sa na jeho vyučovanie v tomto stupni štúdia často používa pohyb postavičky. Opísali sme niekoľko malých programovacích jazykov a edukačných hier, ktoré tento koncept využívajú, a pri opise sme sa zamerali hlavne na použitý typ pohybu a s ním spojenú motiváciu.

V druhej kapitole sme zanalyzovali niekoľko úloh zo súťaže iBobor, ktoré využívali absolútny alebo relatívny pohyb, a následne sme navrhli vlastnú sadu úloh. Testovanie prebehlo v papierovej podobe so žiakmi piatych ročníkov. Zistili sme, že labyrinty nie sú vhodnou motiváciou na relatívny pohyb, a taktiež, že tento typ pohybu je pre žiakov náročnejší. Tieto poznatky nám pomohli pri návrhu sady úloh v prostredí GEVIN, ktorú sme následne otestovali so žiakmi tretieho ročníka – úlohy aj výsledky testovania sme opísali v tretej kapitole.

Žiaci sa s programovaním postavičky pomocou kartičiek stretli po prvýkrát, ale všetci vyriešili pripravených šestnásť úloh. Osem úloh sme venovali absolútnemu pohybu a osem relatívnemu pohybu, pričom štyri úlohy pre každý pohyb boli s okamžitým vyhodnocovaním a ďalšie štyri s vyhodnocovaním až po spustení pripravenej postupnosti príkazov. Zbieranie predmetov sa ukázalo ako vhodná motivácia pre relatívny pohyb, a taktiež sa osvedčila nami zvolená postupnosť úloh. Prvé štyri úlohy v každej časti pomohli žiakom pochopiť pohyb postavičky, prvá úloha, v ktorej sa nevyužívalo okamžité vyhodnocovanie, im zas pomohla rozplánovať si riešenie a zvyšné úlohy už riešili rýchlejšie. Ukázalo sa však, že niektorí žiaci by potrebovali viac úloh zameraných na postupné získavanie vedomostí. Na druhej strane, ďalšia skupina žiakov považovala úlohy za ľahké, vyriešili ich rýchlo a dokázali by vyriešiť aj ťažšie úlohy.

Veríme, že výsledky výskumov, ktoré sme prezentovali v našej práci, môžu pomôcť pri príprave úloh pre žiakov prvého stupňa a zároveň nami navrhnutá postupnosť úloh môže slúžiť učiteľom ako pomôcka na zoznámenie žiakov s programovaním.

## Bibliografia

- [1] Štátny pedagogický ústav, „Štátny vzdelávací program, Informatická výchova,“ Bratislava, 2008.
- [2] Štátny pedagogický ústav, „Rámcový učebný plán pre základné školy s vyučovacím jazykom slovenským,“ Bratislava, 2011.
- [3] Štátny pedagogický ústav, „Inovovaný štátny vzdelávací program, Informatika,“ Bratislava, 2014.
- [4] Štátny pedagogický ústav, „Rámcový učebný plán pre ZŠ s vyučovacím jazykom slovenským,“ Bratislava, 2015.
- [5] S. Lutz a W. Huitt, „Connecting cognitive development and constructivism: Implications from theory for instruction and assessment,“ *Constructivism in the Human Sciences*, zv. 9, %1. vyd.1, pp. 67-90, 2004.
- [6] D. R. Krathwohl, „A Revision of Bloom's Taxonomy: An Overview,“ *Theory into practice*, zv. 41, %1. vyd.4, 2002.
- [7] R. E. Mayer, „Rote versus Meaningful Learning,“ *Theory Into Practice*, zv. 41, %1. vyd.4, pp. 226-232, Autumn 2002.
- [8] K. Pauchly, „Výučbové programovacie mini-jazyky,“ [Online]. Dostupné na: <http://www.flatulent.szm.com/karel/minijazyky.html>. [Cit. 25. Január 2016].
- [9] Baltík, „Baltík 3 - základné informácie,“ [Online]. Dostupné na: [http://baltik.infovek.sk/zakl\\_inf.htm](http://baltik.infovek.sk/zakl_inf.htm). [Cit. 2. Apríl 2016].
- [10] Ľ. Salanci, „EasyLogo,“ [Online]. Dostupné na: <http://www.salanci.sk/EasyLogo/index.html>. [Cit. 25. Január 2016].
- [11] DevTech Research Group, Lifelong Kindergarten, Playful Invention Company, „About ScratchJr,“ [Online]. Dostupné na: <http://www.scratchjr.org/about.html>. [Cit. 25. Január 2016].



- [12] BotLogic.us, „BotLogic,“ [Online]. Dostupné na: <http://botlogic.us/>. [Cit. 28. Január 2016].
- [13] Lightbot, „How does Lightbot teach programming?,“ [Online]. Dostupné na: <https://lightbot.com/hoclearn.html>. [Cit. 29. Január 2016].
- [14] Katedra základov a vyučovania informatiky, FMFI UK, „iBobor,“ [Online]. Dostupné na: <http://ibobor.sk/>. [Cit. 1. Február 2016].
- [15] L. Budinská, „Absolútny a relatívny pohyb vo vyučovaní programovania,“ rev. *Študentská vedecká konferencia FMFI UK, Zborník príspevkov [na publikovanie]*, Bratislava, 2016.
- [16] L. Budinská, *Systém na generovanie úloh na vyučovanie základov programovania (bakalárska práca)*, Bratislava: Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, 2014.
- [17] B. Coudoin, „GCompris,“ 2000-2016. [Online]. Dostupné na: <http://gcompris.net/index-sk.html>. [Cit. 25. Máj 2016].

## **Prílohy**

Digitálna príloha obsahuje:

- 1) Pracovný list
- 2) Žiacku časť aplikácie GEVIN s pripravenou sadou úloh