

Univerzita Hradec Králové
Přírodovědecká fakulta
Katedra kybernetiky

Výuka algoritmického myšlení založená na
programovacím jazyce SCRATCH

Bakalářská práce

Autor: Michaela Hrdinová
Studijní program: B1101 Matematika
Studijní obor: Matematika se zaměřením na vzdělávání
Informatika se zaměřením na vzdělávání
Vedoucí práce: PhDr. Musílek Michal, Ph.D.

Univerzita Hradec Králové

Přírodovědecká fakulta

Zadání bakalářské práce

Autor: Michaela Hrdinová

Studijní program: B1101 Matematika

Studijní obor: Matematika se zaměřením na vzdělávání

Informatika se zaměřením na vzdělávání

Název práce: Výuka algoritmického myšlení založená na programovacím jazyce SCRATCH

Název práce Aj: Teaching of algorithmic thinking based on SCRATCH programming language

Cíl a metody práce: Cílem bakalářské práce je připravit a alespoň zčásti pilotně ověřit možnost výuky algoritmického myšlení a základů programování na vybrané základní škole. Žáci by se měli naučit pracovat se sekvencemi příkazů, cykly a podmínkami a vytvářet jednoduché procedury či funkce s parametry (vlastní bloky). Teoretická část práce zmapuje stávající situaci v České republice a v zahraničí rešeršně kompilační metodou. Hlavním přínosem bude vytvoření pracovních listů s jednoduchými algoritmickými úlohami, odpovídajícími zvolené věkové kategorii žáků základní školy. Vybrané pracovní listy budou následně ověřeny ve výuce a jejich přínos k rozvoji algoritmického myšlení bude ověřen např. zadáním pre-testu a post-testu experimentální a kontrolní skupině žáků a posouzením efektivity pracovních listů a v nich obsažených úloh metodou normalizovaného zisku.

Garantující pracoviště: katedra kybernetiky, Přírodovědecká fakulta UHK

Vedoucí práce: PhDr. Michal Musílek, Ph.D.

Oponent: Ing. Petr Voborník, Ph.D.

Prohlášení:

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a že jsem v seznamu použité literatury uvedla všechny prameny, ze kterých jsem vycházela.

V Hradci Králové dne

Michaela Hrdinová

Poděkování

Mé poděkování patří PhDr. Michalu Musílkovi, Ph.D. za odborné vedení, trpělivost, ochotu a čas, který mi v průběhu zpracování bakalářské práce věnoval.

Anotace

HRDINOVÁ, Michaela. Výuka algoritmického myšlení založená na programovacím jazyce SCRATCH. Hradec Králové, 2018. Bakalářská práce na Přírodovědecké fakultě Univerzity Hradec Králové. Vedoucí bakalářské práce Michal Musílek. 70 s.

Bakalářská práce připravila a z části také ověřila možnost výuky algoritmického myšlení a základů programování na víceletém gymnáziu, na němž autorka již působí v pozici učitele informatiky. Cílem vytvoření pracovních materiálů je, aby se žáci naučili pracovat se sekvencemi příkazů, cykly a podmínkami a dokázali vytvářet jednoduché procedury či funkce s parametry. Teoretická část práce mapuje stávající situaci v České republice a v zahraničí rešeršně kompilační metodou. Praktická část představuje soubor pracovních listů s jednoduchými algoritmickými úlohami, odpovídajícími věku studentů Prímy osmiletého gymnázia. Vybrané pracovní listy byly ověřeny ve výuce a jejich přínos k rozvoji algoritmického myšlení byl zmapován prostřednictvím kontrolních testů.

Klíčová slova:

programovací jazyk, Scratch, programování, algoritmizace, hodnocení, výuka

Anotation

HRDINOVÁ, Michaela. Teaching of algorithmic thinking based on SCRATCH programming language. Hradec Králové, 2018. Bachelor Thesis at Faculty of Science University of Hradec Králové. Thesis supervisor Michal Musílek. 70 p.

This bachelor thesis prepares and (in some parts) verifies ways of teaching algorithmic thinking and basics of programming at high school, where the author works as an IT teacher. The aim of the creation of the working materials is for pupils to start using sequences of orders, series and conditions and to be able to create simple procedures or functions with parameters. The theoretical part maps current situation in the Czech Republic and abroad by search and compilation method. The practical part presents several worksheets with easy algorithmic tasks, which are suitable for approximately twelve-year-old pupils. Some of the worksheets were verified during school lessons and their positives for development of algorithmic thinking were checked via tests.

Keywords:

programming language, Scratch, programming, algorithmic, evaluation, teaching

Obsah

Úvod	9
1 Scratch	10
1.1 Historie.....	10
1.1.1 Mitchel Resnick.....	14
2 Programátorské prostředí	15
2.1 Základní rozdělení plochy.....	15
2.1.1 Scéna	15
2.1.2 Blok pro přidávání postav	16
2.1.3 Blok s příkazy.....	16
2.1.4 Volná plocha pro skládání příkazů	16
2.1.5 Lišta úprav.....	16
2.1.6 Blok s příkazy pro tvorbu scénáře.....	16
2.2 Kostýmy	17
2.3 Zvuky.....	17
3 Zkušenosti z českých a zahraničních škol	18
3.1 Česká republika.....	18
3.1.1 Základní škola Velké Bílovice.....	18
3.1.2 Biskupské gymnázium, Žďár nad Sázavou	18
3.2 Zahraničí.....	19
3.2.1 Harvey Mudd College	19
3.2.2 Locks Heath Junior School	19
4 Další jazyky vhodné pro výuku programování pro děti	20
4.1 Hour of Code.....	20
4.2 Snap! (BYOB = Build Your Own Blocks).....	20
4.3 Alice	21
5 Zahraniční články o Scratch	22
5.1 Procedia – Social and Behavioral Sciences	22
5.2 Opensource.com	22
5.3 African Journal of Research in Mathematics, Science and Technology Education.....	23

5.4	Medium	23
6	Praktická část	25
6.1	Badatelsky orientované vyučování	25
6.2	Hodnocení žáků	27
6.3	Přípravy	28
6.4	Druhý ročník čtyřletého gymnázia vs. kroužek v Primě osmiletého gymnázia.....	58
	Závěr	60
	Zdroje	61
	Přílohy.....	64
	Příloha č. 1 - Presentace k 9. hodině	64

Úvod

S grafickým programovacím jazykem Scratch jsem se poprvé setkala na střední škole, kterou jsem navštěvovala. Bylo to ve druhém ročníku 4-letého gymnázia, kde se tento jazyk učí dodnes. Překvapilo mne, jak jednoduché může programování být. Celé vývojové prostředí je dobře přehledné a má intuitivní ovládání. Na vysoké škole jsme dále měli předmět Programování, který byl zaměřen na dětské programovací jazyky. Tady se Scratch objevil podruhé a já se o něj začala zajímat o trochu více.

Mým cílem při psaní této práce, je rozšířit povědomí o tomto jazyce a o možnosti výuky algoritmizace u dětí 2. stupně základní školy. Také chci poukázat na to, že naučit se základům programování není již tak složité, jako tomu bývalo dříve, a to právě díky programu Scratch.

V první kapitole se budu zabývat historií a vývojem programovacího jazyka Scratch. Tím, kdo jej vytvořil, komu je jazyk Scratch určen a na jakém principu funguje.

Ve druhé kapitole popíšu celé programátorské prostředí jazyka Scratch. Najdeme zde obrázky pro lepší vizualizaci.

Třetí kapitola se zaměří na možnosti výuky algoritmizace v České republice i v zahraničí.

Čtvrtá kapitola stručně zmapuje další alternativy pro výuku programování a algoritmizace u dětí. Uvedu zde přednosti daného programu i způsoby využití.

Pro porovnání situace v České republice se světem v páté kapitole jsem si vybrala 4 různé zahraniční články, které se zabývají jazykem Scratch. Dva články jsou spíše studií, při kterých se pracovalo s nějakou experimentální skupinou a další dva jsou zpovědí, vycházející z osobních zkušeností.

Poslední kapitola, tedy šestá, bude praktickou částí mé práce. V začátku této kapitoly ještě představím formu výuky, kterou jsem ve své učitelské práci využívala nejčastěji. Zároveň sem zařadím podkapitulu s názvem Hodnocení. Hlavním prvkem praktické části bude soubor vytvořených příprav na jednotlivé hodiny. Pomocí těchto příprav lze vyučovat základy programování např. na 2. stupni základní školy. V poslední části kapitoly se zamyslím nad rozdíly ve výuce u různě starých žáků. Konkrétně ve 2. ročníku čtyřletého gymnázia a Primě osmiletého gymnázia.

1 Scratch

Scratch je grafický programovací jazyk určený především pro mládež ve věku od osmi do šestnácti let. Používají ho ale všechny věkové kategorie, bez ohledu na pohlaví či národnost. Nyní se rozšířil do více než 150 zemí světa a je lokalizován do více než 40 jazyků, jak uvádí (Lifelong Kindergarten Group, 2015). Nejčastěji bývá zařazen do výuky na druhém stupni základní školy nebo na středních školách. Je dobré zařadit tento jazyk na začátek výuky programování. Výuka v tomto programu se vyznačuje jednoduchým ovládáním. S barevnými dílky, které představují jednotlivé prvky, se pracuje stylem drag and drop, tj. posunutím myši na správné místo a povolením stisku tlačítka myši. Název Scratch vznikl podle techniky zvané „scratching“, což je pojem, který používají hudební diskžokejové (wikipedia team, 2018) Jedná se o pohyb deskou dopředu a dozadu, čímž skládají jednotlivé hudební prvky do sebe a na základě toho vznikají skladby. Podobně je tomu tak i u programu Scratch, kdy spojujeme jednotlivé díly do sebe a vytváříme program. Nejde zde o žádné psaní kódů, čímž se předchází syntakticky chybným zápisům. Všechny sestavené algoritmy lze zkusit v průběhu celého programování. Máme možnost vidět, jak a co reaguje. Pro děti má Scratch přínos především v podpoře logického myšlení, schopnosti samostatně přemýšlet a zkusit nové věci, porozumět algoritmizaci a schopnost později porozumět složitějším jazykům.

Za pomoci programu Scratch můžeme vytvářet animace, interaktivní hry, různé příběhy nebo například prezentace. Může to posloužit jak dětem, tak i učitelům, kteří si v něm mohou připravit různé materiály pro podporu výuky. Má využití v mnoha předmětech, např. chemii, historii, matematice či cizích jazycích.

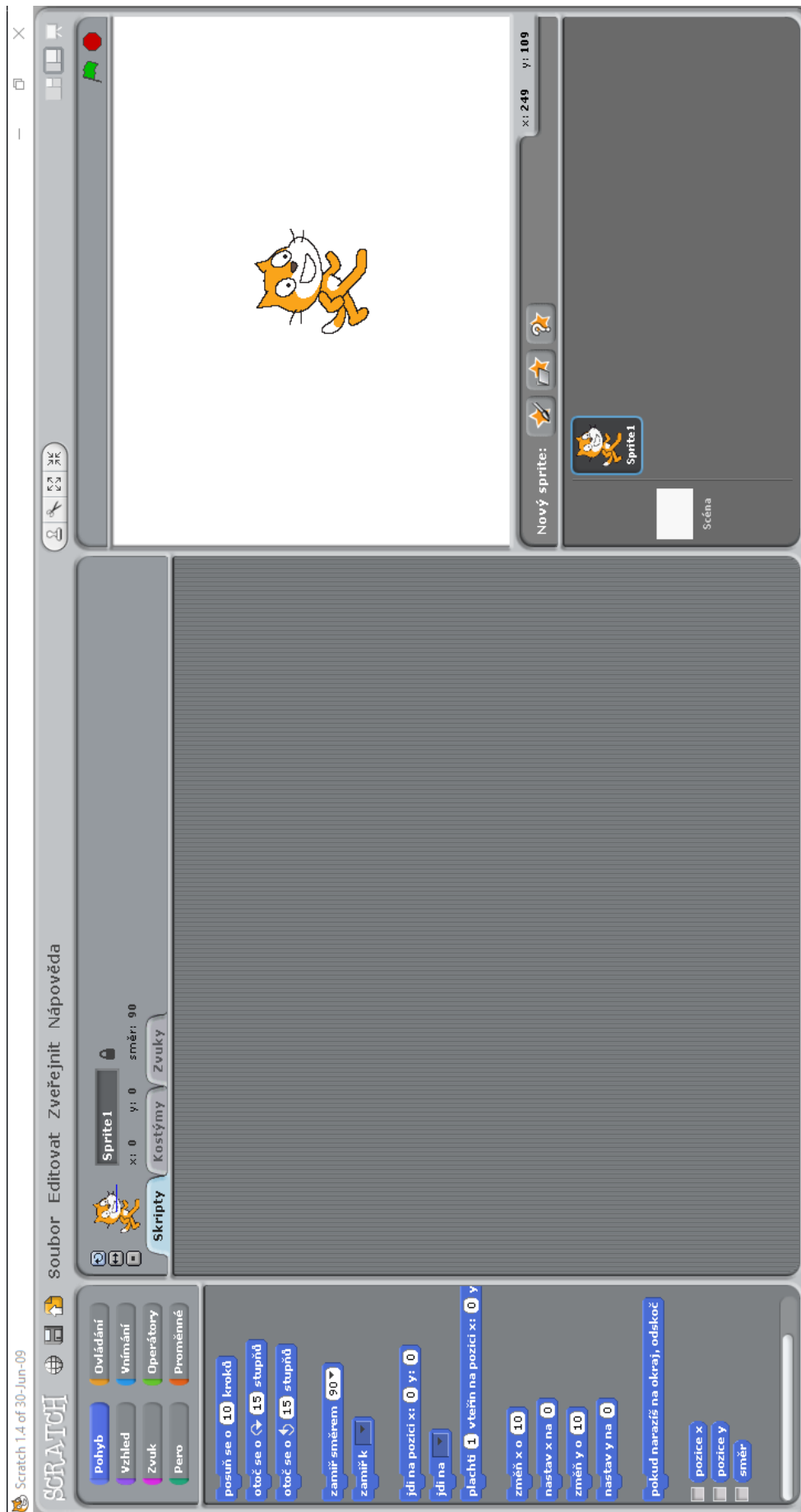
Program je k dispozici zdarma, online na domovských stránkách <https://scratch.mit.edu/>, ale i ke stažení ve verzi 1.4 nebo 2.0.

1.1 Historie

Scratch 1.0 se začal vyvíjet v roce 2003 a byl hotov asi v roce 2007, kdy byly zároveň spuštěny webové stránky, na nichž se daly ukládat sdílet vytvořené projekty. Stabilní verze, tedy 1.4 vznikla roku 2009 a používá se dodnes. Poslední verzí je Scratch 2.0 z roku 2013 (wikipedia team, 2018). Obě tyto verze lze nyní používat, avšak starší Scratch neumí otevírat projekty vytvořené v novější verzi, tedy verzi 2.0.

Dalším rozdílem mezi verzí 1.4 a 2.0 je například to, že novější Scratch má možnost vytvářet bloky, které vlastně představují podprogramy s parametry a které nám umožňují rozdělit celý algoritmus na menší celky, čímž se stává přehlednějším (viz Obrázek 3) Pokud bychom se blíže podívali, našli bychom ještě pár rozdílů, především v jiném uspořádání příkazů, v grafice programu nebo chybějících příkazech „Startuji jako klon“, „klonuj sebe“ a „zruš klon“ u starší verze programu.

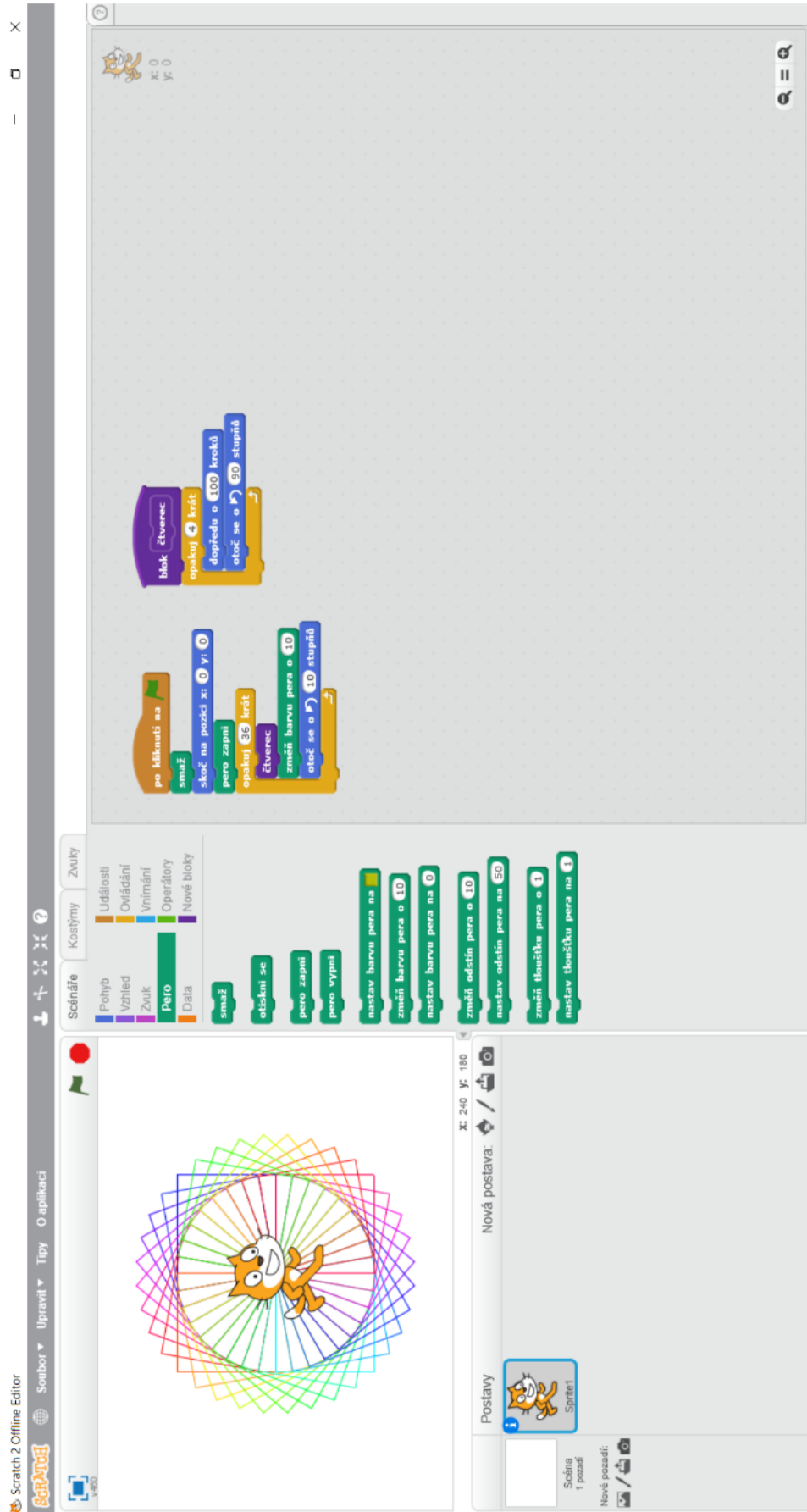
Vytvoření programu Scratch je financováno Národní vědeckou nadací (National Science Foundation) Spojených států amerických. Vytvořila ho skupina Lifelong Kindergarten Group (V překladu Celoživotní mateřská školka) v čele s Mitchelem Resnickem. Skupina působí v univerzitní laboratoři Media Lab Massachusettského technologického institutu (MIT).



Obrázek 1 - Scratch 1.4



Obrázek 2 - Scratch 2.0



Obrázek 3 - bloky

1.1.1 Mitchel Resnick

LEGO Papert Professor of Learning Research

Associate Academic Head, Program in Media Arts and Sciences

Resnick získal titul BA na Princetonovské Univerzitě (1978) z oboru fyzika, dále tituly MS a PhD na MIT (1988, 1992) z oboru počítačová věda. Pracoval jako vědecko-technologický novinář v letech 1978 až 1983. Je autorem knihy „Turtles, Termites, and Traffic Jams“ (1994) a spoluautorem knih „Adventures in Modeling“ (2001) a „The Official ScratchJr Book“ (2015). Získal cenu McGraw v oblasti vzdělávání (2011) a cenu AACE EdMedia Pioneer (2013).

Resnick a jeho skupina Lifelong Kindergarten spolupracují se společností LEGO a snaží se vytvářet nové produkty na podporu výuky. Například LEGO Mindstorms a WeDo, což učí děti vytvářet své vlastní roboty, fungující motory a mnoho dalšího. Pomáhá to rozvíjet jejich dovednosti v oblasti vědy a techniky. (wikipedia team, 2018)

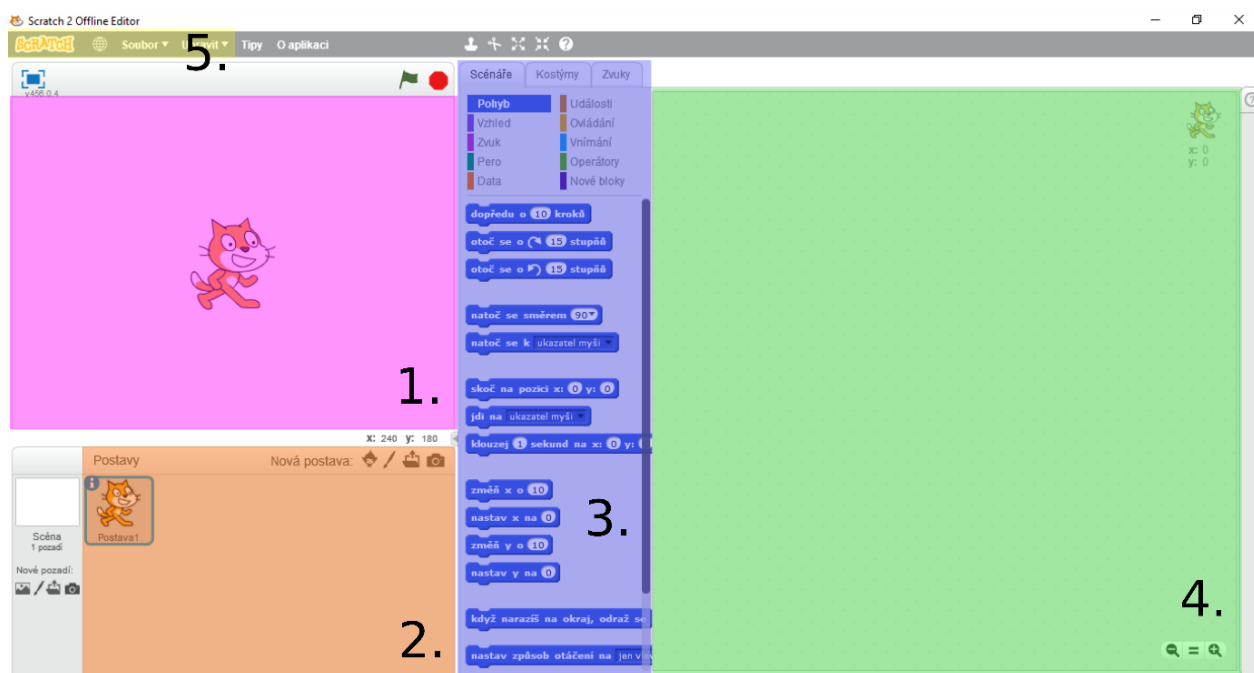
2 Programátorské prostředí

Vývojové prostředí Scratch představuje plnohodnotný nástroj pro tvorbu programů s řadou možností. V dalším textu se zaměřím pouze na ty, které mají výrazný didaktický potenciál vzhledem k cíli mé práce, tj. výuce základů algoritmizace.

2.1 Základní rozdělení plochy

Pracovní plocha je rozdělena na několik celků, které slouží k různým účelům a usnadňují žákům orientaci. Jsou to následující části:

1. Scéna
2. Blok pro přidávání postav
3. Blok s příkazy
4. Volná plocha pro skládání příkazů
5. Lišta úprav



Obrázek 4 – Základní rozdělení plochy

2.1.1 Scéna

Část na levé straně obrazovky (viz Obrázek 1). Po spuštění programu se na scéně objevuje jediný objekt, a to kocour Scratchy. Jedná se o prostor, kde se vizuálně zobrazuje náš vytvořený program. Zde vidíme výsledky naší práce a je to zároveň část, která se zobrazí uživateli, který si vytvořený projekt prohlíží.

Scénáře můžeme vytvářet jak pro postavy, tak i pro Scénu. Scéna má však omezené možnosti, nemá například příkazy na kartě Pohybu.

2.1.2 Blok pro přidávání postav

Část vlevo dole. Zde můžeme přidávat, odebrat, skrývat či kopírovat jednotlivé postavy neboli Sprity. Slovo Sprite v angličtině znamená nadpřirozenou bytost, skřítku nebo vílu. Pro nás je ale Sprite postava, jejíž „chování“ a vzhled můžeme ovlivnit tím jak ji naprogramujeme. Jakmile si otevřeme nový soubor, zobrazí se nám kočka (Scratchy), pokud by se nám však nelíbila, lze ji nahradit nebo doplnit řadou dalších objektů. V knihovně programu jsou zvířata, pohádkové postavy, kreslení nebo reální lidé, ale i věci, rostliny atd. Postavy můžeme přidávat buď z předem vytvořené knihovny, můžeme si je nakreslit, nahrát ze souboru nebo vyfotit přes kameru. Každé postavě poté můžeme vytvářet samostatný scénář.

2.1.3 Blok s příkazy

Sloupec mezi scénou a plochou pro programování. Máme zde tři záložky: Scénáře, Kostýmy a Zvuky. Ve složce Scénáře se nachází jednotlivé příkazy rozřazené do různých podskupin podle použití. Tato složka slouží k vytváření scénářů. Další záložkou jsou Kostýmy, kde přiřazujeme jednomu Spritu různé kostýmy, které může v průběhu scénáře střídat (více v kapitole 2.3. Kostýmy). Poslední záložkou jsou Zvuky, které přiřazujeme Spritům, pokud chceme, aby vydávaly zvuky.

2.1.4 Volná plocha pro skládání příkazů

Pokud zůstaneme na záložce Scénáře, objeví se nám na **pravé straně obrazovky** volná plocha. Nabízí se nám velký prostor pro pokládání jednotlivých dílků scénáře. Na toto místo tedy skládáme náš program. Celé to funguje na principu skládání kostiček lega. Jednotlivé příkazy, dílky, skládáme do sebe tak, aby zapadaly. Program nám v podstatě neumožní udělat chybu v návaznosti příkazů.

2.1.5 Lišta úprav

V samotné hlavičce programu. Po kliknutí na logo Scratch v levé části lišty, se dostaneme zpět na domovskou stránku (<https://scratch.mit.edu/>). Obrázek globusu nám umožňuje změnit jazyk programu. Při rozkliknutí možnosti Soubor, si můžeme zvolit jednu ze čtyř možností: Nový, Nahrát z tvého počítače, Stáhnout do tvého počítače nebo Vrátit. Buď tedy vytvoříme nový projekt, nebo si otevřeme již vytvořený projekt, uložený v počítači. Stejně tak si můžeme uložit rozpracovaný či hotový projekt do svého počítače. A poslední možností můžeme vymazat vše, co jsme vytvořili od otevření projektu.

2.1.6 Blok s příkazy pro tvorbu scénáře

Máme na výběr několik příkazů, které jsou přehledně rozřazeny do podkategorií označených barvami. Podkategorie jsou: Pohyb, Vzhled, Zvuk, Pero, Data, Události, Ovládání, Vnímání, Operátory a Bloky.

2.2 Kostýmy

Na této záložce lze přirozeně měnit kostýmy jednotlivým postavám. Většina postav, které jsou v knihovně, mají v databázi více než jeden kostým. Pomocí změny kostýmů, můžeme nasimulovat reálný pohyb, jako chůzi či let. Stejně jako můžeme postavu nakreslit, můžeme nakreslit i její kostým, nebo nějakým způsobem upravit stávající podobu kostýmu. Při úpravách lze využít buď vektorový režim, nebo převést obrázek na bitmapu, záleží na konkrétním použití.

Kostýmy lze následně do programu zařadit pomocí příkazů na kartě Vzhled. Konkrétně pomocí příkazů „změň kostým na“ a „další kostým“.

2.3 Zvuky

Zvuky můžeme nahrávat z knihovny, z mikrofону nebo ze souboru. Možnosti úprav nejsou nijak rozsáhlé, avšak kdo chce použít kvalitní zvukové efekty, stejně většinou použije speciální editor pro hudbu.

I zvuky mají svoje vlastní příkazy, a to konkrétně na kartě Zvuk. Je zde spousta možností jak zvuk během stavby programu využít.

3 Zkušenosti z českých a zahraničních škol

3.1 Česká republika

Výuka ve formě kroužku:

- ZŠ Velké Bílovice
- ZŠ Holzova 1, Brno
- Biskupské gymnázium, Žďár nad Sázavou

Výuka jako součást tematického plánu:

- Biskupské gymnázium, Žďár nad Sázavou

Ostatní:

- SŠ Obchodní akademie SOVA
- ZŠ Rýmařov

3.1.1 Základní škola Velké Bílovice

Na základní škole ve Velkých Bílovicích funguje tzv. Tech Class, kterou vytvořila skupinka tamních učitelů. Jde o příležitost pro studenty, kteří chtějí proniknout do světa IT, do světa techniky a informatiky. Celý kolektiv se opakovaně účastní školních projektů se zaměřením na rozvoj schopností programování.

Dlouhodobým cílem této skupiny je vytvoření komunitní sítě škol, které mají zájem o technologie a algoritmizace. Celý program je zaměřen především na základní školy, které chtějí kroužek dále rozvíjet.

Kurzy, které kroužek nabízí, jsou: LEGO WeDo, LEGO Mindstorms NXT 1, LEGO Mindstorms NXT 2, LEGO Mindstorms EV3 a Scratch.

3.1.2 Biskupské gymnázium, Žďár nad Sázavou

Výuka programu Scratch na této škole je běžně zařazena do druhého ročníku čtyřletého studia. Dříve než započne výuka v programu, žáci se seznamují s vývojovými diagramy, které by jim měli pomoci lépe pochopit algoritmizaci. Dále se studenti seznamují se samotným programem, to trvá asi 4 měsíce. Nakonec je výuka zakončena rozsáhlejším projektem, který by měl odrážet jejich schopnosti programování.

Minulý rok se nově otevřel i kroužek, který byl pod mým vedením, zaměřen na studenty 6. a 7. třídy.

3.2 Zahraničí

3.2.1 Harvey Mudd College

Profesorka Colleen Lewis, pocházející z této univerzity, vytvořila internetový kurz pro výuku programování v jazyce Scratch. Nezáleží na věku, ani na zkušenostech s programováním, přihlásit se může prakticky každý. Instruktorka si bere za cíl představit veřejnosti programování, které snadno pochopí a budou moci využít. Účastníci kurzu se naučí vytvářet interaktivní příběhy, hry a animace, které budou moci sdílet online. Kurzy jsou koncipovány jako video tutoriály, ve kterých vás profesorka Lewis přesně instruuje co a jak máte dělat. Samozřejmě v angličtině. Kurzy jsou po 6ti hodinách týdně a trvají 6 týdnů. Abyste se mohli vzdělávat, musíte zaplatit členský poplatek, který činí \$49 USD. (LEWIS, Colleen, 2018)

3.2.2 Locks Heath Junior School

Na této základní škole, která se nachází v Anglii, se programování učí děti ve věku od 9 do 11 let. Studentům se rozvíjí jejich znalosti v oblasti algoritmizaci a logického myšlení. Žáci během roku pracují na dvou hlavních projektech. Prvním z nich je „Crab Maze“ neboli „Krabí bludiště“. Jedná se o hru, kde hráč ovládá kraba a má za úkol dovést ho ven z bludiště. Dalším projektem je „Angle Sorter“ neboli „Třídíč úhlů“. Tento program má rozřadit geometrické útvary pomocí jejich vnitřních úhlů do čtyř kategorií:

- Tvary, které obsahují ostré úhly
- Tvary, které obsahují tupé úhly
- Tvary, které obsahují pravé úhly
- Tvary, které mají jiný úhel než výše jmenované – tím může být například kruh

Může se stát, že některý z útvarů bude patřit do více kategorií. Dejme tomu, že budeme mít mnohoúhelník, který obsahuje jak ostré, tak i tupé úhly. Potom bude na nás, do které kategorie ho zařadíme, jestli do první, nebo do druhé.

Díky těmto úkolům žáci získají schopnost rozložit jednotlivé problémy na části a řešit je pomocí jednoduchých algoritmů. Mohou potom sami bez problémů vyřešit i těžší projekty. (Locks Heath Junior School, 2018)

4 Další jazyky vhodné pro výuku programování pro děti

4.1 Hour of Code

V českém jazyce „Hodina kódu“ je celosvětová akce, které se účastní desítky milionů žáků z více než 180 zemí světa. (Code.org, 2015)

Cílem kampaně je poskytnout Hodinu kódu žákům v jednom týdnu 7. – 13. prosince, kdy se slaví Týden informatiky. Není to však podmínkou, je možné si program spustit kdykoliv jindy.

Tento skvělý nápad přišel od veřejné neziskové organizace Code.org. Tato skupina lidí se věnuje zvyšování zájmu o informatiku ve společnosti, zvyšování účasti žen a menšin tím, že ji nabízí k dispozici na více školách. Projekt funguje také díky podpoře velmi dobře známých firem a slavných osobností. Nejznámějšími podporovateli jsou například Microsoft, Apple, Amazon, bývalý prezident Barack Obama, Mark Zuckerberg a další.

Na stránkách Hour of Code si může kdokoliv a kdekoliv spustit výukový program v délce trvání jedné hodiny. Žádné předchozí zkušenosti s programováním nejsou vyžadovány, program vás to všechno naučí sám. Jsou zde kurzy, ke kterým je vyžadován počítač, k některým stačí například jen tablet a některé dokonce nevyžadují počítač vůbec. (Code.org, 2015)

4.2 Snap! (BYOB = Build Your Own Blocks)

Snap! je dalším programovacím jazykem, vhodným pro výuku dětí. Byl představen Kalifornskou univerzitou v Berkeley. Jeho tvůrcem je Jens Mönig, který je součástí společnosti MioSoft. O grafické rozhraní se stará Brian Harvey.

Jedná se o grafický programovací jazyk, který je založen, stejně jako Scratch, na systému drag-and-drop. Je v podstatě rozšířenou verzí programu Scratch. Na rozdíl od něj, umožňuje vytvářet vlastní bloky a používat rekurzi. Díky těmto vylepšením, se nyní může s klidným svědomím použít jako úvod do programování na střední nebo i vysoké škole.

Snap! lze používat online v internetovém prohlížeči, kde běží pomocí Javascriptu. Aktuální verze je Snap! 4.1.2.7. Program je navržen tak, aby neomezoval chod vašeho počítače a vy jste mohli bez obav využívat i projekty dalších uživatelů. Užitečné je, že umožňuje importovat projekty z programu Scratch 2.0. a dále s nimi pracovat. (wikipedia team, 2018)

4.3 Alice

Alice je jedním z dalších programovacích jazyků, který je stejně jako Scratch založen na práci s dílky a bloky. Byl navržen americkým profesorem Randy Pauschem (*23. října 1960 – †25. července 2008), který se aktivně podílel i na jeho tvorbě. Celý software je dále pod záštitou Univerzity Carnegie Mellon.

Tento programovací jazyk není vhodný pro úplné začátky s programováním, zvolila bych ho spíše jako pokračovací nástroj. V tomto programu můžeme vytvářet animace, interaktivní příběhy nebo programovat jednoduché hry ve 3D. Alice je program, který umožňuje studentům lépe pochopit princip objektově orientovaného programování. Zároveň by se dal použít jako úvod do programování v jazyce Java.

V současné době lze najít dvě volně dostupné verze programu – Alice 2 nebo Alice 3. Novější verze, tedy Alice 3, se od své starší verze liší tím, že klade důraz právě na objektově orientované programování. Je zde také možnost zobrazit si celý program v Java kódu, který se generuje sám podle toho, jak sestavujeme algoritmus. Další výhodou je, že projekt můžeme exportovat do platformy NetBeans a následně jej upravovat přímo v jazyce Java. To nám umožní rozšířit některé funkční vlastnosti, které bychom v Alici nezvládli. (Alice, 2017)

Program bohužel v praxi vyzkoušený nemám, ale ráda bych ho někdy využila např. jako pokračovací kroužek po úvodu ve Scratch.

5 Zahraniční články o Scratch

5.1 Procedia – Social and Behavioral Sciences

„Learning Basic Programming Concepts By Creating Games With Scratch Programming Environment“

Článek se zabývá problémem vyučování základů algoritmizace a programování na středních školách v Maroku. Přesněji se jedná o studii, kterou vytvořili Ibrahim Ouahbi, Fatiha Kaddari, Hassane Darhmaoui, Abdelrhani Elachqar a Soufiane Lahmine.

Předmětem této studie bylo ověření hypotézy: „Vytvoření jednoduchých her v programech určených pro začátečníky by mohlo zvýšit motivaci k učení se programování.“

69 Studentů vytvářelo jednoduché hry za použití programu Scratch. Tyto studenti byly náhodně vybráni a rozděleni do tří podskupin. První skupina byla testována pomocí metody založené na vytváření jednoduchých her v programu Scratch. Další dvě skupiny byly vyučovány klasickou metodou s využitím programovacího jazyka Pascal. Na začátku i na konci experimentu se porovnávalo jaké programovací schopnosti žáci mají, jaké jsou jejich herní návyky, jejich motivace a zájem o programování v budoucnosti. Když byly na konci celkového testování žáci tázáni, zda by chtěli pokračovat ve studiu programování, až 65 % z těch, kteří pracovali ve Scratch, uvedlo, že o tom uvažuje. Naopak jen 10,3 % studentů, kteří programovali v jazyce Pascal, projevilo zájem o další vzdělávání v této oblasti.

5.2 Opensource.com

„Is Scratch today like the Logo of the '80s for teaching kids to code?“

Článek je v podstatě založen na osobní zkušenosti autora, kterým je Anderson Silva. V úvodu se zabývá programem Logo, který jeho samotného přivedl k programování začátkem 80. let. Popisuje jak program vypadal, že hlavním objektem zde byla želvička, a jakým způsobem se ovládal. Postupem času se v tomto programu naučil pracovat s cykly, podmínkami, proměnnými i různými procedurami. Jednou z výhod také bylo, že se naučil základy geometrie a kartézské soustavy. Po několika letech, když se snažil probudit zájem o želvičku i ve svých dětech, zjistil, že pro ně tento program není příliš atraktivní. Hledal proto programovací jazyk, který by je zaujal více. A našel Scratch.

Scratch má spoustu společného s Logem, co se týče funkčnosti. Avšak je o něco vyspělejší. Uživatelé si například nemusí pamatovat jednotlivé příkazy, protože ve

Scratch jsou již vytvořeny. Pro děti je lepší v tom ohledu, že je nabádá skládat jednotlivé dílky jako puzzle a vytvořit tak kód.

Na konci článku popisuje jaké jsou aktuální verze programu Scratch. Zároveň přikládá odkazy na program Logo, Scratch a společnost Code.org.

5.3 African Journal of Research in Mathematics, Science and Technology Education

„Lessons Learned from Teaching Scratch as an Introduction to Object-oriented Programming in Delphi“

Tento článek byl napsán a ověřen na Univerzitě North-West, ve městě Potchefstroom, v jižní Africe. Jeho autorkami jsou Sukie van Zyl, Elsa Mentz a Marietjie Havenga.

Jedná se o studii, při které se ověřovalo zavedení programu Scratch do vzdělávacích osnov. Mělo jít o úvod do programování, který by studentům pomohl lépe porozumět programovacímu jazyku Delphi. Zjišťovalo se také jaké zkušenosti získali učitelé během učení programu Scratch v prvním roce. Účelem celého výzkumu bylo zjistit jak může být Scratch efektivně použit jako přechod k jazyku Delphi. Zároveň se snažili o vytvoření učebního plánu, který měl přiblížit úvod do objektově orientovaného programování v jazyce Delphi.

Pro tento výzkum byla vybrána kvalitativní metoda v podobě rozhovorů, které se uskutečnily s 8 náhodně vybranými učiteli Informačních technologií. Cílem bylo, nahlédnout do zkušeností, které učitelé získali během toho prvního roku, kdy byl Scratch zařazen do výuky. Hlavním problémem se ukázalo, že učitelé byli hodně nejistí. A to konkrétně v tom, že nevěděli jak spojit řešení problému a vzhled algoritmu. Tedy jak efektivně sestavit algoritmus, aby byl co nejkratší a zároveň naprosto funkční.

Výsledkem celé studie tedy byly jakési pokyny pro výuku programování v jazyce Scratch, které kladou důraz především na to, aby učitelé uměli sestavit algoritmus, který bude efektivní a srozumitelný. Dále se však zaměřuje na to, aby se studenti uměli vypořádat s různými chybami, naučili se je řešit a následně program otestovat.

5.4 Medium

„How I Think about Scratch and Computer Science“

Sarah Otts, vývojářka online komunity programu Scratch, napsala článek, který vychází převážně z jejích, ale i cizích zkušeností. Ona a zbytek jejího týmu, se

neustále potýká s nedůvěřivými otázkami rodičů a některých pedagogů. Je Scratch vůbec o programování? Proč se studenti neučí radši v JavaScriptu? Naučí Scratch mladé lidi pracovat s počítači?

V úvodu se autorka rozepisuje o vlastní cestě k tomuto programu. Dostala se k němu v 8. ročníku, když hledala programovací jazyk, který by ji bavil více, než ten o kterém se učili ve škole. Nalezla Scratch. Během pár minut se seznámila s uživatelským prostředím a rychle stvořila jednoduchou interaktivní animaci. Její zájem o práci s kódy narostl ve chvíli, kdy nemusela řešit žádné chybové hlášky nebo problémy se syntaxí. Scratch jí inspiroval k rozšíření znalostí o další různé programovací jazyky. Nakonec ji to pohltilo natolik, že se rozhodla jít na vysokou školu. Nyní, po 9 letech prvního kontaktu se Scratch, odpromovala v oblasti počítačových věd. Zvládá spoustu programovacích jazyků a přesto se stále ráda vrací k začátkům. V současné době, pracuje v týmu, který se stará o vývoj online komunity okolo programu Scratch.

Hodně vyučujících, kteří mají s tímto programem zkušenosti a využili jej v praxi, se shoduje na tom, Scratch rozhodně může být dobrým začátkem pro nezkušené budoucí programátory. Claire Pritchard, která pracuje s dětmi na škole v Chicagu, používá ve výuce Scratch a ScratchJr. Děti se v hodinách naučily jak řešit problém a jak pracovat s kódy. Profesorky Mona Rizvi a Thorna Humphries (Norfolk State University) se také zaměřují na využití programu Scratch jako začátek programování. Vedou na univerzitě úvodní kurz k programování, kde využívají právě tento jazyk. Při závěrečném hodnocení, které získaly od studentů, se jim dostalo velmi kladných odpovědí. Jeden student například uvedl, že mu to pomohlo v rozvíjení programovacího myšlení, naučil se dělat věci krok za krokem. Další žák řekl, že tvoření a používání cyklů mu pomohlo v pochopení jazyka C++, který je čekal v navazujícím kurzu.

„Náš konečný cíl je pomoci všem dětem, aby rozvíjeli své myšlení, rozvíjeli své hlasy a rozvíjeli své identity.“ (Mitchel Resnick, 2016)

V tomto programu nejde jen o programování. Dalším cílem je dát lidem každého věku nástroj pro vytváření projektů, které je budou zajímat. Mladí lidé používají Scratch, aby vymýšleli projekty o módě nebo o jídle. Užívají ho, aby vytvářeli hudbu a vlastní animace. Během toho se učí základní práce s počítačem, procvičují schopnost pracovat ve skupině, kreslí a navrhují, konstruktivně přemýšlejí a komunikují.

6 Praktická část

Pro praktickou část mé bakalářské práce jsem si vybrala střední školu, kterou jsem navštěvovala, a na které v současné době působím jako učitelka. Jedná se o Biskupské gymnázium ve Žďáru nad Sázavou. Tato škola má dvě možnosti studia, a to buď osmileté, nebo čtyřleté studium. Ve školním roce 2017/2018 jsem zde učila pouze informatiku a to v rozsahu 13 hodin týdně, z nichž jedna hodina byl Multimediální kroužek. V následujícím roce bych měla učit již 19 hodin týdně a opět pouze informatiku.

Po domluvě s Mgr. Simonou Kadlčíkovou (vyučující informatiky, fyziky a matematiky) mi byl panem ředitelem Ing. Jiřím Cočevem, schválen kroužek zaměřený právě na výuku algoritmizace a programování v programu Scratch. Kroužek je určen pro žáky Primy a Sekundy, což věkem odpovídá 6. a 7. ročníku základní školy. Jde o to, naučit děti logickému myšlení, rozvíjet jejich kreativitu a naučit je samostatně přemýšlet a řešit daný problém. Postupně by se měli naučit, jak pracovat s cykly, podmínkami, proměnnými, zprávami či bloky.

Již při přípravě koncepce kroužku jsem si vytyčila za cíl pozorovat, jak k tomu jednotlivé děti přistupují a jak se zdokonalují. Asi od poloviny celkové časové dotace by žáci měli řešit vhodně zadaný dlouhodobější projekt, na kterém by využili a procvičili získané znalosti (hra, delší příběh).

Aby byla výuka pro děti zajímavější a aby si toho co nejvíce zapamatovaly, rozhodla jsem se zahrnout tzv. Badatelsky orientovanou výuku (viz kapitola 6.1). Myslím, že pro děti je určitě přínosnější než klasická frontální výuka, kde stojí celá hodina pouze na mé aktivitě.

6.1 Badatelsky orientované vyučování

Tato forma výuky se snaží podpořit konstruktivistický styl výuky, podněcuje chuť žáků zkoumat a učit se. Dalším přínosem také je, že se zaměřuje na dobré vztahy a spolupráci ve třídě. U jednotlivých studentů vede k rozvíjení logického myšlení, aktivitě a samostatnému bádání. Během výuky se využívá aktivizujících metod, jakými jsou například kritické myšlení nebo projektová výuka.

Celé to v podstatě funguje tak, že žáci si sami kladou otázky, následně formulují hypotézy, plánují postupy ověření hypotéz, provádějí pokusy a nakonec vyhodnocují výsledky a vytvářejí závěry.

Metoda by se dala rozdělit do 4 kroků:

1. Co chci řešit – motivace, získávání informací, kladení otázek, výběr výzkumné otázky
2. Přicházím s domněnkou – formulace hypotézy
3. Jak zjistím, zda mám pravdu – plánování a příprava pokusu, provedení pokusu, zaznamenávání pokusu, vyhodnocení dat

4. Na konci cesty sklízím ovoce své práce – formulace závěrů, návrat k hypotéze, hledání souvislostí, prezentace, kladení nových otázek

V České republice existuje projekt Badatele.cz, který vznikl na základě spolupráce vzdělávacího centra TEREZA a týmu učitelů z celé republiky. Tato skupina usiluje o zařazení Badatelsky orientovaného vyučování do výuky na základních školách. Na jejich internetových stránkách je možné najít některé ukázkové materiály, jak taková výuka probíhá. Stejně tak tu lze najít názory učitelů i žáků na takto vedené hodiny.

6.2 Hodnocení žáků

Přestože se jedná o dobrovolné hodiny v rámci školního kroužku, to znamená, že známky se studentům nedávají, bylo by vhodné stanovit nějaký systém hodnocení. I když se bude jednat pouze o slovní hodnocení nebo hodnocení ve formě odměny, děti to správně namotivuje a bude je to nutit k lepším výsledkům. Děti, které budou nějakým způsobem pochváleny, budou mít radost z dobře odvedené práce a ty ostatní naopak budou chtít zažít stejný pocit jako jejich spolužáci.

Ohodnocení se tedy bude skládat především z ústní pochvaly. Podle mě je dobré pochválit i ty, kterým to třeba tolik nejde, i když vyzdvihnu jen nějakou maličkost, děti to potěší a budou se snažit být ještě lepší. Jednu hodinu mám naplánovanou jako soutěžní a na konci, až dojde k vyhodnocení výsledků, dostanou žáci odměnu ve formě sladkostí.

Pokus bych měla popsat o jaký typ hodnocení se jedná z hlediska terminologie uváděné v rámci odborné literatury, bylo by to nejspíš Formativní hodnocení.

„Formativní hodnocení má umožnit cílenou radu, vedení a poučení zaměřené na zlepšení výkonů. Odhalování chyb a nedostatků v práci žáka má smysl, když je východiskem k jejich odstranění. Plní především diagnostickou funkci, poskytuje zpětnou vazbu učiteli i žákům, pro řízení učení žáků je nepostradatelné.“

„Snahou není určit, kdo umí nejlépe a kdo nejhůře, ale jak to udělat, aby uměli všichni dobře.“ (Zdeněk Kalhous, Otto Obst, 2002)

Geoff Petty ve své knize Moderní vyučování navrhuje tento postup:

- *„Objasňovat žákům věci tak dlouho, jak budou pokládat za potřebné, případně i individuálně;*
- *Poskytovat jim tolik praxe, kolik pokládají za potřebné;*
- *Upřesňovat, co je nezbytné pro úspěšné zvládnutí úkolu, aby mohlo být procvičování soustředěno správným směrem;*
- *Oznamovat žákům, proč neuspěli, aby si mohli doplnit mezery, opravit chyby či zdokonalit se v dovednostech, v nichž nedosahovali požadovaného standardu;*
- *Umožňovat libovolný počet pokusů.“ (Geoff Petty, 2013)*

Problém tohoto postupu je však v jeho časové náročnosti. Konkrétně v mém případě, kde se jedná o kroužek, bude těžké věnovat se každému zvlášť. Hodinová dotace bohužel není příliš vysoká, proto se zaměřím spíše na tu praktickou část.

Nedostatky a chyby, které se u dětí vyskytnou, budeme diskutovat dohromady pomocí diskuze tak, aby se z chyb poučili všichni.

6.3 Přípravy

Tato kapitola obsahuje přípravy na jednotlivé hodiny kroužku. Přípravy jsem tvořila z hodiny na hodinu, aby na sebe vše navazovalo. Na začátek jsem zařadila ještě přípravu na hodinu v rámci Objevitelského dne, který se na naší škole konal. Byla to úplně první hodina, kterou jsem učila sama a ve které jsem se pokusila naučit děti programovací jazyk Scratch tak, aby si sami dokázali vytvořit jednoduchou hru.

U některých příprav jsem zařadila tzv. Hodnocení. Pokusila jsem se takto zhodnotit průběh dané hodiny, tedy jestli se podařilo vše, co jsem si naplánovala. Tuto část jsem zařadila pouze u hodin, které byly něčím zajímavé, takže např. první hodina, testovací hodina, nebo soutěžní hodina.

Objevitelský den

1. Seznámení s prostředím
 - a. Přepnutí češtiny, ukládání projektů
 - b. Postavy – přidávání z knihovny / kreslením, kostýmy, zvuky
 - c. Pozadí – nastavení obrázku pozadí, kreslení pozadí
 - d. Pole pro skládání příkazů – plocha, kam se přenáší jednotlivé bloky
 - e. jednotlivé kategorie příkazů (stručně a jen ty potřebné):
 - i. pohyb – posuň se o, otoč se o, natoč se směrem, skoč na pozici
 - ii. vzhled (kostýmy) – říkej, změň kostým, nastav velikost
 - iii. zvuky (karta zvuky) – přehraj zvuk
 - iv. pero
 - v. události – začátek každého programu, pro nás buď „po kliknutí“, nebo „po stisku klávesy“
 - vi. ovládání – opakuj dokud, když, zastav vše
 - vii. vnímání – klávesa stisknuta, dotýká se barvy
 - viii. operátory (základní) - +, -, *, / ... použijeme pouze „nebo“
2. Ukázka jednoduchých, základních příkazů – společně, zobrazím na plátně + každý bude zkoušet sám
 - a. Dojít z jednoho okraje na druhý
 - b. Nakreslit čtverec a říct např. „to je krásný čtverec“
 - c. Jít dokud se nedotkne červené čáry a poté přehrát libovolný zvuk
3. Projekt – hra „Projdi bludiště“
 - a. Na e-mailu scratch.je.super@seznam.cz si stáhnout pozadí bludiště
 - b. Nejdříve si ukážeme jak rozpohybovat postavu pomocí šipek – určíme jednotlivé podmínky pro stisk klávesy – musí se to opakovat pořád dokola, abychom mohli šipky stále používat

- i. Nejdříve zkusíme dát opakuj dokola, šipky fungují, ale postava má povoleno dotýkat se všeho
 - c. Zajistíme, aby se postava nesměla dotknout stěn, tzn. postava se ovládá šipkami, dokud se nedotkne stěny – změníme cyklus na „opakuj, dokud nenastane dotýká se barvy černé“, což je barva stěn bludiště
 - i. Co se stane, když se dotkne stěny? Např. něco řekne nebo vydá zvuk, potom se vrátí na výchozí pozici a natočí se správným směrem
 - d. Na konci postava např. vydá zvuk, nebo něco řekne – postava se ovládá šipkami, dokud se nedotkne konce, tzn. omezíme pohyb – do cyklu přidáme operátor „nebo“ a dotýká se barvy konce
 - i. Když dojde na konec, přehraje zvuk, něco řekne
 - e. Určíme výchozí pozici, na kterou se bude postava vždy vracet vždy po kliknutí na zelenou vlaječku, stejně tak i směr natočení
 - f. Nakonec přidáme rozloučení a cyklus ukončíme
4. Pokud zbyde čas – buď si ukážeme rozšíření v podobě nepřítele nebo počítání bodů, nebo může každý vymyslet svůj „zlepšovák“

Nepřítel – vytvoříme nějakého protivníka, který se bude sám hýbat a naše postava se ho nesmí dotknout

- a. Přidáme novou postavu našeho „nepřítele“
 - i. Určíme mu výchozí pozici a směr natočení
 - ii. Chceme, aby sám chodil z jedné strany na druhou – to se bude opakovat, dokud se ho nedotkne naše ovládaná postava
- b. Přidáme podmínku, že se postava bude ovládat šipkami, dokud se nedotýká protivníka – přidáme ještě jeden operátor „nebo“ a novou podmínku, že se dotýká „nepřítele“
- c. Když se ho dotkne – něco řekne / vydá zvuk a vrátí se na výchozí pozici

Počítání bodů – zařídíme, aby se nám počítalo skóre (pomocí proměnné)

Objevitelská sobota – hodnocení + sebehodnocení

Dne 11. 2. 2017 se konal na Biskupském gymnáziu ve Žďáru nad Sázavou tzv. Objevitelský den. Šlo o den, kdy byl pro děti ze školy a okolí připraven vzdělávací program formou pokusů a cvičení. Převážně bylo vše určeno dětem ve věku od 8 do 13 let. Každý žák si vybíral 2 z 5 vzdělávacích bloků, a to z předmětů fyzika, dějepis, informatika a chemie. Já jsem byla požádána, abych se ujala bloku informatiky, a to konkrétně s tématem „Nauč se programovat“, což se mělo odehrávat v prostředí

programu Scratch. Měla jsem si tedy připravit dvouhodinový obsah výuky pro děti od 3. třídy až po 7. třídu. Zároveň se jednalo o jakési lákadlo na nově otevřený kroužek základů programování ve Scratch.

Za celý den jsem učila dvě skupiny dětí. V první skupině (dopolední) bylo dohromady 12 žáků a mezi nimi byly chlapci i dívky. V druhé skupině (odpolední) bylo žáků jen 7 a všichni chlapci. K dispozici jsem měla dva asistenty, studenty, které mi gymnázium poskytlo. Byla jsem opravdu ráda, že tam byli a pomáhali dětem, když třeba zrovna nevěděli. Oba dva to dobře znali, takže vždy věděli, jak poradit.

Protože bylo jasné, že se s tímto programem nejspíš nikdo z dětí nesetkal, připravila jsem si na úvod i stručné seznámení s prostředím a základními příkazy. Po rychlokurzu jsme si vyzkoušeli pár jednoduchých konstrukcí, jako např. nakreslení čtverce, nebo použití podmínky když – tak. Když jsem se ujistila, že všichni všemu rozumí a vědí, jak pracovat, začali jsme pracovat na jednoduché hře s názvem „bludiště“. Chtěla jsem dětem ukázat základní konstrukce, na kterých potom mohou stavět své vlastní výtvořky. Takže jsme rozpohybovali postavu pomocí šipek, nakreslili si vlastní pozadí, vymezili jsme pohyb postavy, aby se nesměla dotýkat stěn bludiště, a nakonec jsme stihli i rozšíření o nepřítele, který měl nastaven automatický pohyb.

Druhá skupina byla dokonce tak rychlá, že jsme stihli ještě druhé rozšíření, což bylo počítání bodů, tedy kolikrát zvládne projít bludiště. Pořád nám však zbýval čas, tak jsem jim dala prostor, jestli by chtěli ještě něco vyzkoušet. Nakonec se dohodli, že by chtěli udělat další level. Stručně jsem jim tedy vysvětlila, jak funguje posílání zpráv, což zvládli naprosto perfektně.

Obě skupiny spolupracovaly úplně skvěle, s dětmi jsem neměla jediný problém a překvapilo mne, jak rychle se všemu učily. Ještě před začátkem vyučování, kdy jsme čekali, než všichni dojdou, jsem jim dala prostor, aby si zatím sami něco zkoušeli. A opravdu jsem nečekala, že první, s čím si budou hrát, budou kostýmy postav. Sami se předem naučili, jak se přidávají nové postavy, jak se postavám nastavují kostýmy, nebo například jak jednu postavu několikrát nakopíruji. Protože jsem chtěla, aby hodina děti co nejvíce zaujala, pracovali jsme i se zvuky. K dispozici měl každý sluchátka, a tak nebylo překvapením, že právě různé zvuky je hodně zaujaly.

V první skupině, kde byly jak dívky, tak i chlapci, jsem měla možnost sledovat rozdílný přístup k práci. Dívky pracovaly spíše na estetické stránce, jejich cvičné programy byly hravé a barevné. Chlapci zase často používali dopravní prostředky nebo mužské postavy. Nejvíce se mnou komunikovali chlapci v první řadě a také jedna holčička, která chodí ke mně do kroužku. V druhé skupině byli všichni kluci velice komunikativní a hodně si vzájemně pomáhali. Úvodní seznámení s programem jsem tedy vzala hodně stručně a spíše jsme pracovali na vypracování hry. Jen minimálně potřeboval někdo z nich pomoc ode mě nebo od asistentů. Také proto jsme celé zadání stihli zpracovat ve velmi krátkém čase.

Když bych měla zhodnotit svůj výkon, myslím, že se můžu, s klidným svědomím, pochválit. Na začátku jsem byla nervózní, ale jak jsme se s dětmi seznámili, vše to ze mě opadlo a já si to dokonce užívala. Snažila jsem se nikam nepospíchat, aby děti měly čas vše v klidu vstřebat, na začátku jsme si ukázali nějaké jednoduché konstrukce tak, aby pochopily, jak to funguje. Nikdy jsem jim nedala přesný návod, jak co mají udělat, ale snažila jsem se, aby si na vše přišli sami. Dávala jsem jim hodně otázek, které je měly nasměrovat ke správnému řešení a ono to fungovalo. Děti dokázaly na všechno přijít takřka samy. Považuji svoje vyučování za úspěch a jsem sama na sebe pyšná. Po celém dni za mnou přišel jeden student, asistent, který mi řekl, že na moji výuku slyšel velmi dobré reakce a že děti odcházely opravdu nadšené. To mne neuvěřitelně povzbudilo a upevnilo v přesvědčení, že jsem si budoucí povolání učitelky informatiky zvolila správně.

1. hodina

Datum: 10. 2.

Cíl: Žák se orientuje v prostředí programu Scratch, žák sestaví jednoduchou posloupnost příkazů

Téma: Tvorba jednoduchých algoritmů

Obsah: V úvodu bychom si měli ujasnit, co se děti v kroužku naučí a jakým způsobem bude výuka probíhat. Ukážu žákům, kde mohou tento program bezplatně získat a jakým způsobem ho nainstalovat. Také jsem založila dvě nové e-mailové adresy, jednu společnou pro případné sdílení prací mezi sebou. Druhá má sloužit pro zasílání domácích úkolů, to znamená, že přístup na ni mám pouze já.

Po organizačních záležitostech se můžeme přesunout k seznamování se s programem. Začneme malým teoretickým výkladem o tom, kdo je Scratchy a jak se v programu zorientovat. Ukážeme si pole pro skládání příkazů, práci s pozadím a postup při ukládání projektu. Nahlédneme do jednotlivých kategorií příkazů. Samozřejmě nebudu dětem popisovat každý příkaz zvlášť, zdůrazním pouze ty nejdůležitější a zbytek si ukážeme až v průběhu celého kroužku.

Jakmile dokončíme tento úvod, můžeme se přesunout k samotnému programování. Pro začátek bychom mohli zkusit postavu rozpohybovat, tedy nechat jí dojít např. na druhý konec obrazovky. Podle toho, jak nám to půjde a jaký bude čas, zkusíme nakreslit čtverec (zatím bez použití cyklu).

Za domácí úkol budou mít vytvořit jednoduchý algoritmus, který bude určovat nějaký pohyb postavy.

Metody: Úvodní instruktáž, samostatné objevování prvků a funkcí vývojového prostředí

Formy: Samostatná práce, v úvodní části krátká frontální výuka

Pomůcky: dataprojektor, počítač

Hodnocení: Úvodní hodina byla v duchu seznamování se s prostředím a s některými příkazy. Stihli jsme si ukázat, jak se přidávají nové postavy, jak se mění pozadí a kam se skládají jednotlivé příkazy. Potom jsme si akorát vyzkoušeli některé jednoduché konstrukce, jako je dojít z jednoho okraje na druhý nebo nakreslit čtverec. Protože

má kroužek jen 45 minut, víc jsme toho bohužel nestihli, na začátku se řešili ještě některé organizační záležitosti, začali jsme tedy i o chvíli později.

2. hodina

Datum: 17. 2.

Cíl: Žák umí sestavit algoritmus s cyklem, žák umí vytvořit program, který simuluje pohyb postavy

Téma: Cykly a jejich využití

Obsah: Pro začátek si dáme krátké opakování z minulé hodiny. Zopakujeme si, jak přidat novou postavu, jak změnit pozadí a čím musí začínat každý algoritmus, respektive jakým příkazem. Každý zkusí vytvořit krátkou animaci, přičemž bude změněné pozadí i postava.

První řídicí strukturou, kterou se budeme zabývat, jsou cykly. Na jednoduchém příkladu jim zkusím vysvětlit, jak cykly fungují a ukážu jim rozdíl mezi příkazy „opakuj stále“ a „opakuj _ krát“. Jakmile si budu jistá, že všichni všechno pochopili, přejdeme na něco složitějšího. Naprogramujeme netopýra, který bude létat z jedné strany obrazovky na druhou a u toho mávat křídly.

Pokud nám zbyde čas, zkusí si každý vybrat jinou postavu, která obsahuje kostýmy k simulaci pohybu, a naprogramuje ji stejně jako netopýra.

Za domácí úkol budou mít obdobu stávajícího programu. Každý si najde vhodnou postavu, změní pozadí a naprogramují ji tak, aby simulovala pohyb. Oproti programu vytvořeném ve škole, bude místo chůze z jednoho okraje na druhý, opisovat čtverec nebo obdélník.

Další týden mají jarní prázdniny, takže na úkol mají všichni dostatek času. Výsledný program mi mohou poslat na e-mail nebo odevzdat až na další hodině.

Metody: Úvodní opakování, krátká instruktáž, samostatné objevování prvků a funkcí vývojového prostředí

Formy: Samostatná práce, skupinová práce

Pomůcky: dataprojektor, počítač

Hodnocení: Na začátek hodiny jsem zařadila opakování z minulé hodiny. Protože jsem měla o další 2 žáky navíc, nechala jsem ostatní, aby jim to ve zkratce vysvětlili. Řekli jsme si, jak se přidávají nové postavy, jak se mění pozadí, kam se skládají příkazy a jak můžu měnit kostýmy. Nakonec jsem jim zadala, aby každý vytvořil

kratičkou animaci, ve které změní pozadí a nastaví jinou postavu, než je ta výchozí. K mému překvapení to všichni vzali až moc důkladně, a tak jsme s tím zabrali o něco více času, než bylo v plánu. Většinou si všichni hráli se zvukovou stránkou a s kostýmy.

Jako program hodiny jsem měla naplánované cykly. Během hodiny jsme měli vytvořit netopýra, který se bude pohybovat z jedné strany scény na druhou a u toho mávat křídly. Hodiny nemám nastavené tak, že bych jim ukázala postup a chtěla ho po nich zopakovat, ale snažím se je vyučovat metodou „Badatelsky orientované výuky“ (viz kapitola 6.1). Takže jim spíše dávám návody a otázky, které je vedou k zamyšlení se nad problémem. Proto nám možná vyřešení jednoho úkolu trvá o něco déle, než by mělo. Avšak věřím, že tohle jim dá o něco více než jen opisování příkladu z tabule.

Protože další týden mají děti jarní prázdniny, zadala jsem jim domácí úkol. Projekt, který vypracovali v hodině, měli předělat, aby netopýr mával křídly a u toho opisoval čtverec / obdélník. Úkol mi mají poslat na můj e-mail do příští hodiny.

3. Hodina

Datum: 3. 3.

Cíl: Žák umí pracovat s algoritmem s cyklem, žák umí naprogramovat postavu tak, aby se pohybovala náhodně

Téma: Cykly a jejich využití, náhodná čísla

Obsah: V úvodu hodiny bych s dětmi ráda probrala jejich domácí úkoly. Některé projekty mi nepřišly, nejspíš z technických důvodů. Následně bychom si mohli ukázat některá řešení a společně vymyslet to nejjednodušší.

Pro co nejlepší procvičení si zkusí, do projektu z minulé hodiny, přidat chodící lidskou postavu. Celé by to měli zvládnout sami, případně s menší pomocí.

Jako další úkol jsem si pro ně připravila práci s náhodnými čísly. Vytvoříme algoritmus, který bude simulovat náhodný pohyb libovolné postavy. K tomuto algoritmu využijeme nově příkazy „náhodné číslo od _ do _“ a „opakuji stále“. Postavu tedy necháme klouzat na náhodně vygenerované pozice. Po sestavení funkčního algoritmu, bychom mohli zkusit vytvořit jednoduchou hru. Podstatou hry je právě náhodný pohyb postavy a potom už je na hráči, aby postavu, kliknutím na ni, chytil. Tzn. když klikneme na postavu, zastaví se a vydá zvuk nebo něco řekne (pomocí komiksové bubliny). Děti dostanou krátkou nápovědu k řešení a zbytek už bude na nich.

Na konci hodiny také zadám nový domácí úkol, konkrétně vyplnění pracovních listů (viz obr. 5 a obr. 6). Jsou zde úkoly na:

- doplnění chybějících příkazů v algoritmu, tak aby fungoval;
- „uhádnutí“, co se stane po proběhnutí algoritmu;
- jednoduchou orientaci v programu, formou otázek.

Metody: Diskuse vypracovaných úkolů, krátké opakování, instruktáž, samostatné objevování

Formy: Hromadná výuka, samostatná práce, krátká frontální výuka

Pomůcky: dataprojektor, počítač

Hodnocení: Po zahájení hodiny jsme konzultovali domácí úkol z minulé hodiny. Spousta dětí ho neodevzdalo kvůli technickým problémům. Převážně jim nešel

Scratch spustit, což se samozřejmě může stát. Zjistila jsem, že některým se nepodařilo správně nainstalovat program. Poradila jsem jim tedy znovu jak postupovat a v případě, že by se to někomu opět nepodařilo, napsala jsem jim internetový odkaz na online verzi programu. Po vyřešení problému, jsem jim ukázala, jak úkol řešili ti, kterým se podařilo úkol poslat a jak bych úkol vyřešila já. Všechny děti, které úkol odeslaly, jej řešili stejně a to způsobem, který vyplývá ze znalostí získaných v minulé hodině. Způsob je to možná delší, ale rozhodně správný. Poté jsem jim ukázala, jak bych takový úkol řešila já, za použití cyklu a znalosti z první hodiny, kdy jsme si ukazovali, jak se kreslí čtverec (viz obr. 7 a obr. 8)

Na začátek hodiny jsem jim zadala, aby do projektu z minulé hodiny (létající netopýr), přidali ještě lidskou postavu, která bude simulovat chůzi. Ani jsem jim nemusela říkat, jak se to dělá a během 5 minut měli všichni hotovo. Potěšilo mne, že se z nich stávají praví IT odborníci, protože se mě sami ptali, jak se dá algoritmus zkopírovat, aby ho nemuseli celý vytvářet znovu. Chvilku potom, co jsem jim to ukázala, už byly ruce nahoře, že mají hotovo.

Ve druhé části hodiny jsme programovali žábu, která se měla náhodně pohybovat po obrazovce. Vygenerování náhodné pozice žáby nám sice chvilku trvalo, ale nakonec to všichni pochopili a byli šťastní, že jim to funguje. Nějakým způsobem našla většina tlačítko pro klonování postavy, takže asi 5 minut jsme nadšeně řešili, že se všem množí žáby či jiné postavy. Protože nám zbýval čas, chtěla jsem, aby z projektu udělali hru. Postupným zkoušením a po pár nápovědách, jsme se k řešení dostali a děti byli spokojené, že jim jednoduchá hra funguje.

Na konci hodiny jsem jim zadala nový domácí úkol, a to pracovní listy, které mají do příští hodiny odevzdat.

Hodina byla celkově bez problému, děti skvěle spolupracovali, odpovídali na všechny moje otázky a když něco nevěděli, snažili se poradit si mezi sebou.

Pracovní list

K vypracování úkolů můžete použít i samotný program a jednotlivé algoritmy si v něm vyzkoušet.

1. Doplň chybějící místa v kódu. (Napiš jaký příkaz patří na bílé pole)

- a. Postava se posouvá směrem k okraji. Když se dotkne okraje, zastaví se a říká: „Au! Narazil jsem!“



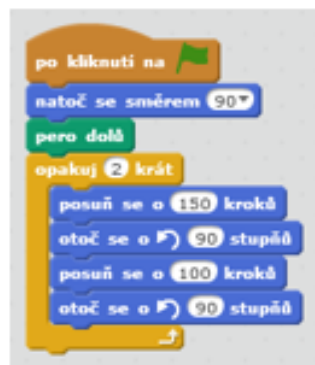
- b. Netopýr se stále dokola pohybuje z jedné strany obrazovky na druhou a u toho mává křídly.



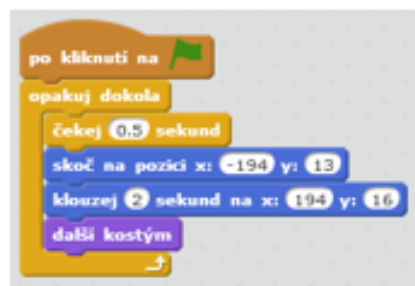
Obrázek 5 - Přední strana pracovního listu

2. Co se stane po proběhnutí kódu?

a.



b.



3. Doplň odpovědi:

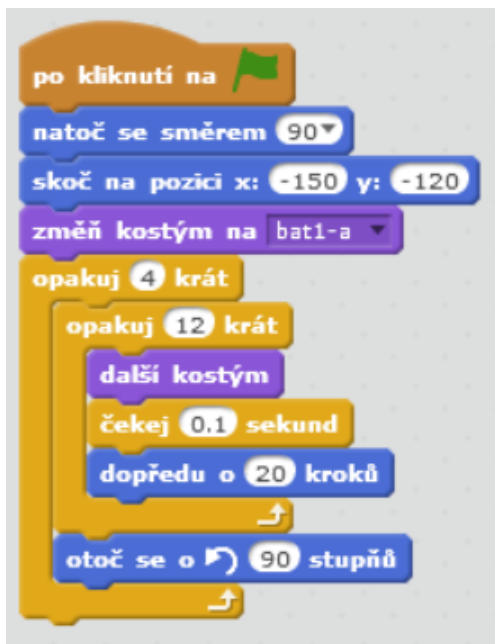
a. Mohu u pozadí použít příkaz z karty POHYB? (modrá karta)

b. Čím začíná každý algoritmus? Kde takový příkaz najdu? (na jaké kartě?)

c. Mohu si nakreslit vlastní postavu? Jaké jsou další možnosti přidání postavy?

d. Jaký příkaz použiju, když chci něco opakovat, dokud se nesplní podmínka? Kde tento příkaz najdu? (na jaké kartě?)

Př. Chci, aby se postava posouvala o 10 kroků, dokud nenarazí na okraj



Obrázek 7 - Řešení moje



Obrázek 8 - Řešení studentů

4. Hodina

Datum: 10. 3.

Cíl: Žák zná všechny druhy cyklů a umí s nimi pracovat, žák umí pracovat s podmínkami

Téma: Cykly, podmínky

Obsah: Ze začátku hodiny společně zkontrolujeme domácí úkol z minulé hodiny. Společně dáme dohromady správná řešení a vysvětlíme si, proč to má zrovna tohle určité řešení.

V této hodině dokončíme cykly. Chybí nám probrat poslední, a to konkrétně příkaz „opakuj, dokud nenastane __“. Aby to děti co nejlépe pochopili, ukážeme si jeho fungování na příkladu. Naprogramujeme postavu tak, že se bude pohybovat až dokud se nedotkne barevného pruhu, který nakreslíme na pozadí, poté změní kostým. U tohoto příkladu si zároveň vyzkoušíme, jak lze vytvořit pozadí pomocí malování.

Cykly máme tímto probrané a rozhodně se k nim budeme ještě dále vracet. Další téma k výuce jsou tzv. podmínky. Pro začátek si představíme jednoduchý podmínkový příkaz „když __ tak“. Jako vždy si jeho funkčnost ukážeme na příkladu. Zadáním příkladu je: „Kočka honí myš, až dokud se jí nedotkne“. Řešení budeme vytvářet spolu a takovým tempem, aby všichni všechno chápali a stíhali.

Metody: Diskuse řešení pracovních listů, společné objevování

Formy: Hromadná výuka

Pomůcky: dataprojektor, počítač, pracovní listy

5. Hodina

Datum: 17. 3.

Cíl: Žák umí načítat data od uživatele, žák ví, co jsou proměnné a k čemu se užívají, umí používat podmínky a sestavit jednoduchou kalkulačku

Téma: Podmínky, proměnné, načítání dat od uživatele

Obsah: Tato hodina bude obsahově trochu náročnější než ty předchozí, ale už vím, jak děti pracují a jsem si jistá, že to všichni zvládnou bez větších problémů.

První věcí, kterou chci dětem ukázat, je načítání dat od uživatele. K tomu použijeme příkazy „ptej se __ a čekej“ a „odpověď“. Necháme postavu, aby se uživatele zeptala, jak se jmenuje, ten následně napíše své jméno a postava ho pozdraví.

Jako další už se podíváme, jak naprogramovat jednoduchou kalkulačku. Abychom pochopili, jak to celé bude fungovat, vytvoříme nejprve algoritmus pro sčítání dvou čísel. Začneme vytvořením dvou proměnných – „cislo1“ a „cislo2“, které je možno žákům připodobnit jako zásuvky, do kterých si ukládáme věci, které budeme později potřebovat. Do těchto proměnných budeme ukládat čísla, která zadá sám uživatel, tzn. použijeme znalosti z úvodu hodiny o načítání dat uživatelem. Nakonec bud' vytvoříme ještě jednu proměnnou „výsledek“, anebo vložíme výpočet rovnou do odpovědi, kterou nám sdělí postava.

Pokud se nevyskytnou žádné problémy, zkusíme kalkulačku o něco vylepšit. Vytvoříme novou proměnnou „volba“. V tuto chvíli využijeme příkaz, který jsme se naučili v minulé hodině, a to „když __ tak“. Když se spustí algoritmus, postava si od uživatele vyžádá zadání čísla 1, 2, 3 nebo 4. Podle toho, jaké číslo uživatel napíše, kalkulačka bude sčítat, odčítat, násobit nebo dělit. Společně si ukážeme řešení pro jednu volbu, zbytek už zkusí vytvořit sami.

Metody: Instruktaž, společné i samostatné objevování

Formy: Hromadná výuka, samostatná práce

Pomůcky: dataprojektor, počítač

6. Hodina

Datum: 24. 3.

Cíl: Žák umí sestavit jednoduchou i složitější kalkulačku

Téma: Podmínky, proměnné, načítání dat od uživatele, operátory

Obsah: V minulé hodině jsme nestihli úplně dokončit kalkulačku, takže v úvodu hodiny nechám děti, aby si práci dodělali a případně se ptali, kdyby jim nebylo něco jasné.

Připravila jsem si ještě jednu kalkulačku na procvičení. Tato kalkulačka by měla umět spočítat obsah a obvod čtverce, obdélníku a kruhu. Nejdříve si ale zopakujeme vzorečky pro výpočty, aby to náhodou neměl někdo špatně jen kvůli chybnému vzorci. Program bude fungovat tak, že postava se uživatele zeptá, který útvar bude chtít spočítat. Po tom, co si uživatel vybere, postava si vyžádá příslušné proměnné. Takže např. u čtverce bude chtít stranu a , u obdélníku stranu a a stranu b . Pomocí operátorů postava spočítá obsah i obvod a sdělí uživateli výsledky.

Nechám děti, aby na řešení této kalkulačky pracovaly buď samostatně nebo klidně ve skupině, ale aby se se mnou radily co nejméně. Proto očekávám, že řešení nám vyjde tak akorát do konce hodiny.

Metody: Krátká instruktáž, samostatné objevování

Formy: Krátká hromadná výuka, samostatná práce

Pomůcky: dataprojektor, počítač

7. Hodina

Datum: 31. 3.

Cíl: Žák umí využít znalosti z předchozích hodin k řešení daného problému

Téma: Cykly, načítání dat od uživatele, proměnné, podmínky, operátory, náhodná čísla

Obsah: Jsme v podstatě v půlce našeho kroužku, proto si chci děti trochu otestovat, abych věděla, jak dokáží využít znalosti, které jsme se během šesti hodin naučili.

Cílem testu je naprogramovat jednu ze dvou variant zadání. Obě varianty jsme někdy v předchozích hodinách zpracovávali, takže by to neměl být až takový problém. Zadání budou mít děti k dispozici, jak v psané formě, tak i v grafické podobě. Pokud by měli problém a nevěděli si rady, jsem ochotná poradit jim, jaké mají použít příkazy.

První variantou je výpočet obsahu čtverce. Zadání:

1. Po kliknutí na postavu, postava nakreslí čtverec, poté se zeptá, jestli chce uživatel spočítat obsah čtverce;
2. pokud bude odpověď ne, postava něco zahlásí a algoritmus skončí;
3. pokud bude odpověď ano, postava se uživatele zeptá na hodnotu strany a (odpověď tedy uloží do proměnné)
4. výpočet se provede až když postava bude říkat výsledek.

Druhá varianta se týká pohybu postav. Zadání:

1. První postava se pohybuje náhodně po celé obrazovce;
2. druhá postava se pohybuje do čtverce / obdélníku a simuluje chůzi / let;
3. po kliknutí na náhodně se pohybující postavu, se vše zastaví (algoritmus se ukončí)

S největší pravděpodobností to dětem zabere většinu hodiny, což nevadí, alespoň si řešení mohou řádně promyslet a nepracovat pod tlakem. Pokud nám přeci jen zbyde trocha času, můžeme si ukázat správné řešení.

Metody: Krátká instruktáž, samostatné přemýšlení

Formy: Krátká frontální výuka, samostatná práce

Pomůcky: dataprojektor, počítač, listy se zadáním projektů

Hodnocení: Když jsem na začátku hodiny, dětem sdělila, že je budu testovat, převládaly ve třídě trochu rozpačité pocity. I když děti ví, že jde jen o kroužek, je vidět, že jim záleží na tom, aby uměly co nejvíce a aby to poté uměly i použít. Chvíli jim trvalo, než si vybraly, které zadání jim více vyhovuje. Viděla jsem, že někteří se prvně pustili do jednoho a v průběhu jim došlo, že neví, jak dál, tak udělali to druhé. Všichni pracovali velice soustředěně, bylo na nich vidět, že nad algoritmem opravdu přemýšlí. Všimla jsem si, že dívky se ze začátku hodně soustředí na to, aby bylo vše graficky doladěné, hrály si s pozadím, měnily kostýmy postav.

Na konci hodiny měli téměř všichni hotovo, až na jednu holčičku, která je zároveň v celé skupině nejmladší. Tato holčička se do kroužku přihlásila, přestože chodí teprve do 4. třídy základní školy a je to dcera kolegyně. Měla trochu problémy s tím, že jí program nefungoval tak, jak by měl. Po hodině jsme to ale spolu daly do pořádku a ona tak mohla odejít s pocitem, že to zvládla.

Určitě je v další hodině všechny pochválím i když to některým nefungovalo tak, jak bych chtěla, všichni se velmi snažili a v mnoha případech vytvářeli opravdu zajímavá řešení. Všichni si zvolili druhé zadání testu, tedy pohyb dvou postav. Jedna holčička byla ale tak rychlá, že stihla řešení i prvního zadání. A musím podotknout, že obě řešení měla správně, což mne velmi těší.

8. Hodina

Datum: 7. 4.

Cíl: Žák umí naprogramovat postavu tak, aby se dala ovládat šipkami

Téma: Jednoduchá hra – podmínky, ovládání šipkami

Obsah: První část hodiny bych chtěla věnovat zhodnocení té předešlé, respektive projektům, které mi děti v rámci testování odevzdaly. Nechám jednoho ze správných řešitelů od každého zadání, aby ostatním vysvětlil, jakým způsobem při programování postupoval. Ukážeme si i špatné řešení a řekneme si kde byly chyby. Chci, aby se děti z těch chyb poučily a aby přesně pochopily jakým způsobem se problém řeší správně.

Druhá část se bude zabývat naprogramováním pohybu postavy, tak aby se dala ovládat klávesami. Nejdříve si zkusíme sestrojít algoritmus pro ovládání postavy šipkami. Je důležité, aby tento postup pochopili, protože se jedná o základ většiny her, které budeme časem vytvářet. Poté zkusíme vytvořit jednoduchou hru, kde se jedna postava bude pohybovat náhodně a druhá postava, kterou ovládá uživatel, ji bude chytat. Když se postavy třikrát dotknou, hra končí. Nakonec do hry přidáme počítání životů nebo bodů (pomocí proměnné).

Metody: Diskuse předešlých projektů, názorná ukázka, krátká instruktáž, samostatné přemýšlení

Formy: Krátká frontální výuka, samostatná i společná práce

Pomůcky: dataprojektor, počítač

9. Hodina

Datum: 21. 4.

Cíl: Žák umí využít získané znalosti k praktickému cvičení bez počítače, žák umí pracovat ve skupině

Téma: Algoritmizační soutěž – tvorba jednoduchých, případně i složitějších algoritmů

Obsah: Pro tuto hodinu jsem si připravila poněkud neobvyklý způsob výuky programování. Děti nebudou používat počítače a budou pracovat ve skupinách. Jedná se o soutěž, při které se vyzkouší jejich logické myšlení, schopnost organizovat činnost ve skupině a práce při časové tísní. V první řadě je nutné děti rozdělit do skupin. Samozřejmě bude záležet na počtu dětí ve třídě, ale chtěla bych, aby v každé skupině byla jedna dívka a aby rozdělení bylo spravedlivé. Následně si ve skupinách vymyslí jméno, kterým je budu nazývat. Skóre jednotlivých skupin bude znázorněno na tabuli, tak, aby děti měli stále přehled jak si vedou. Každá úloha bude ohodnocena jiným počtem bodů.

Během celé hodiny budeme pracovat se speciálními papírovými příkazy, které jsem pro tuto soutěž vytvořila (viz Obrázek 5). Zadání pro jednotlivé úlohy bude vyobrazeno pomocí dataprojektoru a prezentace (viz Přílohy).

Soutěž je rozdělena do tří celků:

1. Základní úlohy

V první řadě se všechny papírové příkazy rozhází na jednom místě učebny. Na začátek jsou připraveny tři základní úlohy, které děti budou muset pomocí papírových příkazů poskládat – Čtverec, Sněhová vločka, Překážka

Hra začne tak, že každá skupina si určí pořadí, ve kterém budou jednotliví členové chodit pro příkazy. Každý člen si může vzít jen jeden příkaz a takto se střídají až dokud nesestaví algoritmus, který splní danou úlohu. V této úloze se nejlépe prověří schopnost komunikace ve skupině. Děti si musejí uvědomit, které příkazy jim chybí k sestavení algoritmu a sdělit si to mezi sebou, aby se nestalo, že někdo přinese příkaz, který je nadbytečný. Skupina, která si bude myslet, že má splněno, přihlásí se a já výsledek zkontroluji.

Každá úloha je jinak bodovaná. Čtverec – 1 bod, Sněhová vločka – 2 body, Překážka – 1 bod. Body dostane skupina, která úlohu vyřeší jako první.

2. Správnost algoritmů

Děti dostanou vytištěné obrázky (stejně jako jsou v prezentaci). Na obrázku je vyobrazené bludiště, kde je nějaká postava, která má dojít do cíle. Postava se pohybuje po čtverečkovaném poli, kde každé pole znamená jeden krok. Děti mají určit správný algoritmus, pomocí kterého postava dojde do konce. Vybírají ze tří algoritmů, kde vždy jeden je správný. Skupina, která odpoví jako první správně, získá 1 bod. Bludiště jsou tu dvě, takže celkem mohou získat 2 body.

3. Ukaž co umíš!

Cílem poslední části soutěže, je sestavit co největší počet smysluplných a funkčních algoritmů. Příkazy děti získávají stejně jako při první části tzn. chodí po jednom a nosí příkazy skupině, která z nich skládá algoritmy. Časový limit není, měly by se rozebrat všechny příkazy a poté uvidím podle nejrychlejší skupiny. Nakonec uděláme zhodnocení jednotlivých algoritmů a rozdělení bodů.

Aby mohl být algoritmus ohodnocen jako správný, musí být především funkční a musí být objasněno, co se stane po jeho proběhnutí. Za každý správný algoritmus, dostane skupina 1 bod.

Na závěr hodiny proběhne vyhodnocení, tedy určení pořadí jednotlivých skupin. Pro všechny mám připravené symbolickou maličkost v podobě sladkosti.

Metody: Krátká instruktáž, skupinové přemýšlení, diskuze nad vytvořenými algoritmy

Formy: Společná práce

Pomůcky: dataprojektor, papírové příkazy, vytištěné obrázky

Hodnocení: Po úvodním zjištění, že nebudeme používat počítače, vypadali děti dost zmateně. Jakmile jsem jim ale řekla, že je čeká soutěž a že na konci dostanou sladkou odměnu, hned zbystřili. Po utvoření skupin se poměrně bavili nad vymyšlením jména skupiny, ale trvalo to o něco déle než bych si přála.

Při první soutěži, kdy měli skládat základní úlohy, se projevila soutěživost některých dětí. Všichni běhali po třídě, a snažili se být co nejrychlejší, občas se ve skupinách překřikovali a chvíli trvalo než se zkoordinovali jako skupina. Všimla jsem si, že holky jsou ve skupinách trochu v pozadí, příliš se neprojevují, což je škoda, protože jsou všechny moc šikovné. Bohužel, kluci mají poněkud silnější hlas a holky většinou překřičí. Také ta soutěživost je u nich zřetelnější.

Ve druhém úkolu se všichni snažili, kluci dávali holkám trochu více prostoru a společně se snažili přijít na správné řešení. Hodně si žáci pomáhali rukama, aby se zorientovali v prostoru.

V posledním úkolu, kde měli vymýšlet co nejvíce algoritmů, jasně vedla jedna skupina. Vymyslela skvělé algoritmy, které byly funkční, přesné a efektivní. Tato skupina tuto úlohu také vyhrála a vím, že to bylo zásluhou především jedné žačky, která většinu těch algoritmů sestavila. Ostatní skupiny se také snažily, i když některé algoritmy jsem nemohla uznat, protože děti nedokázaly vysvětlit co se stane po jejich proběhnutí.

Celkově bych soutěž zhodnotila jako přínosnou jak pro mě, tak i pro děti. Já jsem si vyzkoušela, že děti lze naučit z části programování i bez použití počítače. Děti si vyzkoušely, že práce ve skupině je rychlejší a efektivnější. Spolupracovníci mohou pomoci odhalit chyby a mají zase trochu jiný pohled na daný problém.



Obrázek 9 - Papírové příkazy vytvořené pro hru

10. Hodina

Datum: 28.4.

Cíl: Žák využívá získané dovednosti k tvorbě složitější hry

Téma: Tvorba složitější hry s více úrovněmi

Obsah: V předchozích hodinách jsem dětem sdělila, že posledních několik hodin budeme pracovat na složitějším projektu. Dala jsem jim zároveň na výběr, jakou formu by ten projekt mohl mít. Zda se bude jednat o hru, naučný program nebo třeba interaktivní příběh. Celkem jednohlasně se shodli na tom, že chtějí naprogramovat hru.

Doma jsem si pro děti připravila nějakou přibližnou verzi toho, jak si to představuji. Samozřejmě vytváříme to především tak, aby je to bavilo, a co nejvíce se díky tomu naučily, takže finální verzi nechám na dětech. Měli bychom vytvořit klasickou hru, ve které se postava bude pohybovat skákáním z jednoho místa na druhé, nesmí spadnout na zem a musí se takto dostat až do cíle. Jakmile utvoříme základní šablonu hry, mohou si tam děti přidat cokoliv budou chtít. Společně se tedy v této hodině podíváme na správné naprogramování skákání z místa na místo.

Na začátku hodiny si každý vytvoří své herní prostředí, tzn. vybere postavu, pozadí a nakreslí si jednotlivá políčka, na kterých bude postava skákat (asi 3). Zároveň si každý vytvoří políčko „cíl“, na které se má postava dostat, aby se hra ukončila. Protože vím, že s tímto si děti hodně hrají, zabere to určitě docela podstatnou část naší hodiny. Zkusí si na závěr naprogramovat pohyb postavy pomocí šipek doprava a doleva (šipky nahoru a dolů, nebudeme v této hře potřebovat).

Metody: Názorná ukázka, krátká instruktáž, samostatné přemýšlení

Formy: Krátká frontální výuka, samostatná i společná práce

Pomůcky: dataprojektor, počítač

11. Hodina

Datum: 5.5.

Cíl: Žák využívá získané dovednosti k tvorbě složitější hry

Téma: Tvorba složitější hry s více úrovněmi

Obsah: V minulé hodině jsme začali vytvářet první složitější hru. Všichni žáci stihli vytvořit herní prostředí a většina zvládla navíc naprogramovat pohyb pomocí dvou šipek. Kdo to nestihl, má na začátku prostor na dokončení, abychom začínali všichni ze stejné pozice.

V této hodině budeme chtít naprogramovat skok. Postava bude skákat po stisku klávesy mezerník, a to doprava nebo doleva (podle směru natočení postavy). To znamená, že vždy když postava skočí, posune se o nějakou vzdálenost doprava/doleva. Zároveň nastavíme, aby se při seskoku dolů, zarazila o naše vytvořená políčka, a nepadala tak dolů z obrazovky.

Pokud budeme stíhat, zkusíme vylepšit naše vytvořená políčka. U každého z políček, kromě toho cílového, chceme nastavit jakýsi časový spínač. Bude to fungovat tak, že jakmile se políčka jednou dotkne postava, políčko za 3 sekundy zmizí. Zvýšíme tím celkovou obtížnost hry a zároveň můžeme tuto znalost využít v dalších úrovních.

Metody: Názorná ukázka, krátká instruktáž, samostatné přemýšlení

Formy: Krátká frontální výuka, samostatná i společná práce

Pomůcky: dataprojektor, počítač

12. Hodina

Datum: 12.5.

Cíl: Žák využívá získané dovednosti k tvorbě složitější hry

Téma: Tvorba složitější hry s více úrovněmi

Obsah: Poslední část z minulé hodiny stihli jen ti nejrychlejší, na začátku se tedy budeme věnovat dodělávání, aby všichni všechno chápali a stíhali.

Ti, kteří už mají z minulé hodiny práci hotovou, budou přemýšlet nad tím, jak naprogramovat postavu a jednotlivá políčka tak, aby když na nich postava stojí moc dlouho a políčka už zmizí, postava propadla dolů a hra skončila. Případně mohou pomoci těm, kterým to jde pomaleji a potřebují poradit.

Společně se podíváme na další část. Přidáme nápisy jako „Konec hry“ a např. „Gratuluji“. Samozřejmě každý si tam může napsat co bude chtít. Nápis „Konec hry“ se objeví v případě, že postava dopadne dolů na konec obrazovky. Nápis „Gratuluji“ se objeví, jakmile se postava dotkne cílového políčka. Zkusím dětem jen poradit a potom je nechám samostatně či ve dvojicích přemýšlet. Na závěr si prodiskutujeme jednotlivá řešení.

Poslední, co bychom měli v této hodině stihnout, je vylepšit zobrazování nápisů. Jakmile se nápisy zobrazí, měly by zmizet všechna políčka (pro lepší čitelnost) a klidně i postava. Můžeme také nastavit časovou prodlevu, aby se texty zobrazily až třeba po 1 sekundě.

Za domácí úkol budou děti přemýšlet o tom, co jsem zadávala na začátku hodiny rychlejším žákům. Celkem mají na vypracování 14 dní a své výsledky mi mohou posílat na e-mail.

Metody: Názorná ukázka, krátká instruktáž, samostatné přemýšlení

Formy: Krátká frontální výuka, samostatná i společná práce

Pomůcky: dataprojektor, počítač

13. Hodina

Datum: 26.5.

Cíl: Žák využívá získané dovednosti k tvorbě složitější hry

Téma: Tvorba složitější hry s více úrovněmi

Obsah: Žáci mi během volna posílali výsledky svého snažení o vytvoření funkčního algoritmu. Celkem jsem obdržela 2 plně funkční řešení a jedno z nich se mi líbilo natolik, že jsem se rozhodla ukázat ho i ostatním. Chtěla jsem, aby sama autorka tohoto řešení vysvětlila dětem jak k němu přišla a zkusila jim ukázat jak k tomu dojdou i oni. Takže v podstatě část hodiny nechám za sebe učit jednu žačku. Já budu pouze v pozici pozorovatele a případného rádce.

Ve zbývajícím čase, doděláme posledních pár úprav, před tvorbou další úrovně. Např. cílové políčko se objeví, až postava skočí na poslední políčko před ním. Ukážeme si používání bloků. V tomto případě, vytvoříme blok „pohyb“, který bude obsahovat algoritmus pro ovládání šipkami. Tím se náš hlavní algoritmus celkem zkrátí a bude o něco přehlednější.

Ke konci si zkusíme vysvětlit princip posílání zpráv.

Metody: Názorná ukázka, krátká instruktáž, samostatné přemýšlení

Formy: Krátká frontální výuka, samostatná i společná práce

Pomůcky: dataprojektor, počítač

14. Hodina

Datum: 2.6.

Cíl: Žák využívá získané dovednosti k tvorbě složitější hry

Téma: Tvorba složitější hry s více úrovněmi

Obsah: Jedná se o poslední hodinu, ve které budeme pracovat na této hře. Dokončíme tedy kapitolu posílání zpráv a potom vyřešíme všechny nefunkční části. Většinou se jedná o špatné sestavení kódu, což zapříčinila nejspíš nepozornost.

Dále už bych to nechala na dětech. Každý si zkusí vymyslet další úroveň. Pomocí zpráv propojí obě úrovně. Když postava dojde na konec první úrovně, automaticky přejde na další – objeví se nové pozadí, nová políčka a třeba i nová postava. Úrovně by měly být těžší a těžší a můžeme také zkusit přidat počítání skóre. Nechám děti tedy samostatně pracovat a v případě problémů jim pomůžu.

Metody: Názorná ukázka, krátká instruktáž, samostatné přemýšlení

Formy: Krátká frontální výuka, samostatná i společná práce

Pomůcky: dataprojektor, počítač

15. Hodina

Datum: 9.6.

Cíl: Žák umí pracovat i v jiném programovacím prostředí, než je Scratch

Téma: Hour of code = „Hodina kódu“

Obsah: Poslední hodinu bych chtěla opět netradiční cestou otestovat, co všechno se děti během kroužku naučili a jestli jsou schopní přizpůsobit se i jinému programovacímu prostředí (princip sestavování algoritmů je však stejný).

Na začátku bych dětem řekla něco málo o této kampani, o co se jedná, kde si mohou vše vyzkoušet. Jako první by si vyzkoušeli hodinu Minecraft (Adventurer). Společně si „test“ spustíme a zkusíme se zorientovat v tomto prostředí. Ukážeme si, kam mají děti skládat algoritmus, kde se zobrazuje výsledek a kde je přehled všech úkolů.

U posledního úkolu mají vymyslet vlastní algoritmus. Chtěla bych, aby vytvořili něco nápaditého, u čeho budu vidět, že si na tom opravdu dali záležet. Až algoritmus dokončí, zavolají mě, abych viděla, jak si celou dobu vedli.

Kdo „test“ dokončí, vybere si jinou hodinu a znovu to zkusí celé projít. Už to však nebude součástí testování.

Metody: Krátká instruktáž, samostatné přemýšlení

Formy: Krátká frontální výuka, samostatná práce

Pomůcky: dataprojektor, počítač

Hodnocení: Po otázce, kdo hraje nebo alespoň zná hru Minecraft, zvedla ruku většina třídy a všichni se začali docela těšit na to co přijde. Ani jim nevadilo, že se bude jednat o test. Po spuštění internetové stránky Hour of code, nikdo nevěděl, co má čekat, tak jsem jim trochu přiblížila celý koncept této kampaně a hned jsme se pustili do řešení první hodiny kódu. Slyšela jsem názor, že je to zábavnější než samotný Minecraft a opravdu bylo vidět, že to všechny baví. Děti se snažili řešit vše samostatně, jen si občas kontrolovali, u kterého úkolu jsou zrovna spolužáci.

V průběhu programování jsem slyšela porovnávání i s opravdovou hrou Minecraft. Například proč postava pěstuje obilí, když by mohla pěstovat zeleninu, na jednu stranu je hezké, že nad tím takto přemýšlejí a logicky vše propojují. Někteří během řešení používali pomůcky, jako ukazování rukou, aby věděli kam, a o kolik stupňů se mají pootočit.

U posledního úkolu, kde měli samostatně přemýšlet a sestavit algoritmus, jsem se setkala s kratšími i s delšími kódy. Někdo stavěl mosty, někdo domy, jiný zase zasadil různé plodiny. Dva kluci zase nechali rozestavět nálože TNT ve tvaru smajlíku. Bylo vidět, že si s tím všichni dali práci. Nikdo už nepoužíval několik stejných kódů za sebou, ale raději využili cykly.

Nakonec jsem se zeptala, jak se jim forma této písemné práce zamlouvá a dostalo se mi odpovědi, že je to moc baví a že takové písemné práce by se jim líbily ve všech předmětech.

6.4 Druhý ročník čtyřletého gymnázia vs. kroužek v Primě osmiletého gymnázia

V této kapitole bych se chtěla zabývat Sextou a druhým ročníkem čtyřletého gymnázia, což jsou vlastně dvě souběžné třídy (dále je budu souhrnně označovat jako druhý ročník). Zaměřila jsem se na porovnání dvou různých skupin dětí. Předem jsem si stanovila několik otázek, které považuji za zajímavé z hlediska vyhodnocení.

Jak se liší výuka algoritmizace a programování v jazyce Scratch?

Řekla bych, že se moc neliší od výuky mladších dětí. Vše se musí vzít úplně od začátku tak, aby to pochopili i ti, kteří upřednostňují společenské vědy před před počítačovými vědami.

Kroužek: Viz kapitola 6.3 Přípravy.

Druhý ročník: Jako první je nutné pochopit principy algoritmizace, což se ve druhém ročníku dělá standardně pomocí vývojových diagramů. Několik hodin se proto věnuje jejich tvorbě. Následuje seznámení s programem Scratch. Vše se dělá postupně, tzn. od jednoduchých příkazů, přes cykly, podmínky a početní operace až po práci s bloky a tvorbu složitějších algoritmů. Na konci celé kapitoly mají děti vytvářet závěrečný projekt, který by měl být už trochu rozsáhlejší a odrážet schopnosti, získané během výuky.

Pro koho je složitější danou problematiku pochopit?

Nevím, jestli je zde jednoznačná závislost na věku žáka. Je to spíše o daném dítěti. Někdo logické myšlení má, někdo zase ne. Ale pokud bych přece jen měla určit která věková kategorie je na tom lépe, řekla bych, že ta mladší. Je to třeba proto, že mají „jednodušší myšlení“. A tím nemyslím, že by ty děti byly hloupější, to určitě ne. Jen se nedívají na daný problém tak složitě, jako děti starší. Občas jsem byla překvapená i já sama, že úkol, který jsem dětem zadala, vyřešily daleko jednodušeji než já. Samozřejmě mladší děti to baví o něco více, což je důsledkem toho, že si kroužek zvolili dobrovolně. Tím, že to děti baví, tím se jim to i lépe učí.

Kroužek: Na začátku kroužku bylo tempo hodin trochu pomalejší, než bych chtěla, ale zároveň je mi jasné, že menší děti potřebují trochu více času, aby vše pochopily. Na druhou stranu, jakmile jsme vše probrali, děti pracovaly bez problému ať už samostatně nebo ve dvojicích. Většinou všemu rozuměly a mě už téměř nepotřebovaly.

Druhý ročník: Tady to bylo o něco složitější. Studenti se zde dělí na několik skupin: ti, kteří už někdy programovali nebo programují; ti, kteří nikdy neprogramovali, ale zajímá je to; ti, kteří nikdy neprogramovali nezajímá je to.

Samozřejmě, že pro někoho, kdo chce danou látku pochopit, nebo s ní již přišel do styku, je vcelku lehké učivo pochopit. Horší je to pro ty, kdo se s programováním nikdy nesetkali a pravděpodobně ani nikdy nesetkají. Umět to nechtějí, protože jim to v budoucnu k ničemu nebude, přesto z toho potřebují mít známku. Toto je skupina, pro kterou je tedy asi nejsložitější vše ohledně algoritmizace pochopit.

Koho Scratch více baví?

Kroužek: V tomhle je asi jasná odpověď už od začátku. Menší děti tento program určitě baví více. Myslím, že je pro ně ideální, protože ovládání je velice intuitivní. Stačí dětem ukázat základy práce a oni už si na všechno ostatní přijdou sami. Většinou, co jsem pozorovala, si děti hráli především s grafickou stránkou jejich projektů.

Druhý ročník: Ale musím říct, že i starší žáci si práci v programu Scratch celkem užili. Bylo to vidět především na závěrečných projektech. Spousta projektů byla výborně graficky zpracovaná, nápaditá a především funkční. Jsem si jistá, že někteří nad tím strávili opravdu spoustu času. Pro některé žáky je však tento program příliš primitivní, a proto je nebavilo programovat v něm. Ve většině případů to ale byli kluci, kteří navštěvovali kroužek pokročilejšího programování, kde pracovali v programu Visual Studio a v jazyce C#.

Jaký je rozdíl v hodnocení?

Kroužek: Protože se jednalo o dobrovolný kroužek, hodnocení známkami zde samozřejmě neprobíhalo. Hodnotila jsem děti tedy pouze ústní pochvalou. Snažila jsem se neříkat dětem, že mají něco špatně. Pokaždé když měl někdo nefunkční algoritmus, snažila jsem se ho postrčit k tomu, aby vytvořil správné řešení. Mohli se radit i ve dvojici, ale zároveň jsem kladla důraz na to, aby věděli proč to tak je. V některých hodinách, především v té soutěžní, dostávali žáci pochvalu i v podobě sladkostí.

Druhý ročník: V běžné výuce už musí být použity známky. Já osobně jsem se to snažila udělat tak, aby měli šanci i ti, kteří nemají logické myšlení a programování jim nic neříká. Celou výuku jsem tedy měla rozdělenou do několika menších částí, ze kterých žáci postupně získávali známky. Mají tak menší části na učení. Pokud tedy někdo, nezvládá např. algoritmy s cykly, ale rozumí tomu, jak fungují příkazy s podmínkou, dostane dvě známky, které budou určitě lepší, než kdyby se práce psala jako jeden celek. Na konci všichni musí odevzdávat závěrečný projekt, který má vlastně nejvyšší váhu a zahrnuje vše, co se probralo. Nemyslím si ale, že by to bylo nějak složité. Žáci mají možnost poradit se se mnou i se svými spolužáky, takže je na nich, jak toho využijí.

Závěr

Tato bakalářská práce má oslovit především rodiče, děti a učitele. Má za úkol stručně představit programovací jazyk a demonstrovat, že naučit se algoritmizaci a programování není příliš složité. V programu Scratch se děti naučí vytvářet animace, hry či výukové programy, které se dají využít téměř ve všech učebních předmětech.

Po přečtení práce by měl každý vědět jak vypadá programové prostředí jazyka Scratch a jaký je rozdíl mezi verzí 1.4 a 2.0. Zároveň jsem se snažila ukázat jaké jsou další možnosti výuky programování (Hour of Code, Snap!, Alice), od úplných počátků a seznámení s algoritmizací až po pokročilejší programování. Vše je založeno na stejném principu „drag and drop“, proto si z nich může vybrat každý, bez ohledu na množství znalostí ze světa programování.

V jedné kapitole uvádím, jak je využíván Scratch v České republice a jak v zahraničí. Většina dostupných zdrojů, obsahujících návody k programu Scratch, je stále ještě v anglickém jazyce. Do českého jazyka je toho zatím přeloženo jen málo, možná proto je výuka programování v jazyce Scratch u nás tak málo rozšířená. Učitelé se do výuky nechtějí pouštět, protože k tomu nemají dostatečnou oporu. Tím se zabývám i v kapitole se zahraničními články, kde nalezneme jak vědecké studie, tak i články vycházející z osobních zkušeností.

Praktická část mé práce má sloužit jako předloha, jak by mohla vypadat výuka programovacího jazyka Scratch v rámci nepovinného kroužku. Aby byla představa výuky celistvá, zahrnula jsem i kapitoly o tom jak lze výuku koncipovat, tedy jakou použít metodu a jak nejlépe hodnotit studentskou práci. Měla jsem také možnost porovnat práci u dětí různého věku, což by mohlo pomoci některým učitelům rozhodnout se, ve kterém věku je nejlepší zahrnout výuku programování.

Cíle vytyčené zadáním práce se mi podařilo naplnit.

Zdroje

1. Alice [online]. ©1999-2017 [cit. 2018-05-15]. Dostupné z: <https://www.alice.org/>
2. BADATELÉ.CZ. 4 badatelské kroky. *Badatele.cz* [online]. 2012-2018 [cit. 2018-05-15]. Dostupné z: <http://badatele.cz/cz/4-badatelske-kroky>
3. BIGY. Informační a komunikační technologie. *Bigyvr.cz* [online]. 2012-2018 [cit. 2018-05-15]. Dostupné z: <http://www.bigyvr.cz/studium/informacni-a-komunikacni-technologie>
4. CODE.ORG. Nejčastější dotazy. *Hourofcode.com* [online]. ©2015 [cit. 2018-05-15]. Dostupné z: <https://hourofcode.com/cz#faq>
5. ČECHOVÁ HUMPOLCOVÁ, Tereza. Scratch ve výuce. In: *Spomocnik* [online]. 26.01. 2012 [cit. 2018-05-15]. Dostupné z: <https://spomocnik.rvp.cz/clanek/14905/SCRATCH-VE-VYUCE.html>
6. HALOUSKOVÁ, Alena. *Programování pro děti*. 1. vydání. Brno: Computer Press, 2013. ISBN 978-80-251-3809-0.
7. KALHOUS, Zdeněk, Otto OBST a kol. *Školní didaktika*. 2. vydání. Praha: Portál, 2002, s. 405-407. Kap. 19.2 Typy hodnocení. ISBN 978-80-7367-571-4.
8. LEWIS, Colleen. Programming in Scratch. In: *edx* [online]. ©2012-2018 [cit. 2018-05-15]. Dostupné z: <https://www.edx.org/course/programming-scratch-harveymuddx-cs002x-1>
9. LOCKS HEATH JUNIOR SCHOOL. Year 5 Scratch Projects. *Locksheathjunior.com* [online]. [cit. 2018-05-15]. Dostupné z: <http://www.locksheathjunior.com/year-5-scratch-projects/>
10. MIT MEDIA LAB. Lifelong Kindergarten. *Media.mit.edu* [online]. [cit. 2018-05-15]. Dostupné z: <https://www.media.mit.edu/groups/lifelong-kindergarten/overview/>
11. MIT MEDIA LAB. Mitchel Resnick. *Media.mit.edu* [online]. [cit. 2018-05-15]. Dostupné z: <https://www.media.mit.edu/people/mres/overview/>

12. OTTS, Sarah. How I Think about Scratch and Computer Science. In: *Medium* [online]. Dec 05, 2016 [cit. 2018-05-15]. Dostupné z: <https://medium.com/scratchteam-blog/how-i-think-about-scratch-and-computer-science-b376111a5df>
13. OUAHBI, Ibrahim, Fatiha KADDARI, Hassane DARHMAOUI, Abdelrhani ELACHQAR a Soufiane LAHMINE. Learning Basic Programming Concepts By Creating Games With Scratch Programming Environment. In: *ScienceDirect* [online]. 2014 [cit. 2018-05-15]. Dostupné z: <https://sciencedirect.com>
14. PETTY, Geoff. *Moderní vyučování*. 6. vydání. Praha: Portál, 2013, s. 345. Kap. 38. Hodnocení. ISBN 978-80-262-0367-4.
15. *Scratch* [online]. [cit. 2018-05-15]. Dostupné z: <https://scratch.mit.edu/>
16. *Scratch: programování pro děti* [online]. ©2018 [cit. 2018-05-15]. Dostupné z: <http://scratch.chaputo.cz/>
17. SILVA, Anderson. Is Scratch today like the Logo of the '80s for teaching kids to code?. In: *Opensource.com* [online]. Mar 29, 2017 [cit. 2018-05-15]. Dostupné z: <https://opensource.com/article/17/3/logo-scratch-teach-programming-kids>
18. *Snap!: Build Your Own Blocks* [online]. [cit. 2018-05-15]. Dostupné z: <https://snap.berkeley.edu/>
19. ŠKOLA HOLZOVA. Programovací jazyk – Scratch. *Zsholzova.cz* [online]. [cit. 2018-05-15]. Dostupné z: http://www.zsholzova.cz/informatika_programovani.php
20. VAN ZYL, Sukie, Elsa MENTZ a Marietjie HAVENGA. Lessons Learned from Teaching Scratch as an Introduction to Object-oriented Programming in Delphi. In: *Taylor&Francis Online* [online]. Jun 07, 2016 [cit. 2018-05-15]. Dostupné z: <https://www.tandfonline.com/doi/abs/10.1080/18117295.2016.1189215>
21. Wikipedia Team. Mitchel Resnick. In: *Wikipedia* [online]. July 8, 2018 [cit. 2018-07-30]. Dostupné z: https://en.wikipedia.org/wiki/Mitchel_Resnick
22. Wikipedia Team. Scratch (programming language). In: *Wikipedia* [online]. July 29, 2018 [cit. 2018-07-30]. Dostupné z: [https://en.wikipedia.org/wiki/Scratch_\(programming_language\)](https://en.wikipedia.org/wiki/Scratch_(programming_language))

23. Wikipedia Team. Snap! (programming language). In: *Wikipedia* [online]. July 8, 2018 [cit. 2018-07-30]. Dostupné z: [https://en.wikipedia.org/wiki/Snap!_\(programming_language\)](https://en.wikipedia.org/wiki/Snap!_(programming_language))
24. ZŠ RÝMAŘOV. Výuka 1.st. *Zs1.zsrymarov.cz* [online]. ©2018 [cit. 2018-05-15]. Dostupné z: <http://zs1.zsrymarov.cz:8080/index.php/vyukove-materialy/10-vyuka-1-st/13-informatika-1-st>

Přílohy

Příloha č. 1 - Prezentace k 9. hodině



Poskládejte algoritmus
podle zadání

PROCVIČOVÁNÍ

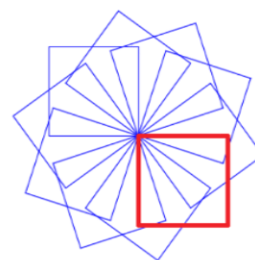
ČTVEREC

- Za pomoci cyklu s opakováním sestrojte čtverec.

PROCVIČOVÁNÍ

SNĚHOVÁ VLOČKA

- Za pomoci **10ti čtverců**, bloků „**opakuj** _ **krát**“ a „**otoč se o** _ **stupňů**“ sestroj sněhovou vločku. Mezi všemi čtverci bude úhel **36°**.



ŘEŠENÍ

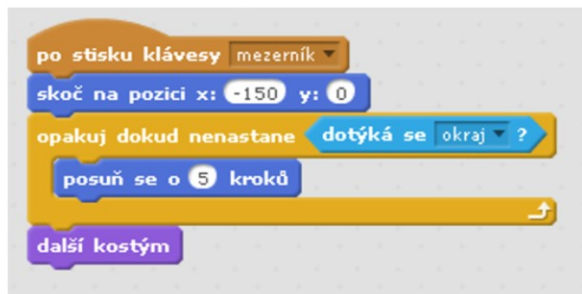


PROCVIČOVÁNÍ

PŘEKÁŽKA

- Po stisknutí klávesy mezerník jdi, až dokud nenarazíš na okraj hrací plochy, potom změň kostým.

ŘEŠENÍ



Který algoritmus je
správný?

Pomozte
kočičce dostat
se ke své
koblize!



A)

```

po kliknutí na [ ]
opakuji 3 krát
  posuň se o 1 kroků
otoč se o 90 stupňů
posuň se o 1 kroků
otoč se o 90 stupňů
posuň se o 2 kroků
otoč se o 90 stupňů
posuň se o 2 kroků
  
```

B)

```

po kliknutí na [ ]
posuň se o 3 kroků
otoč se o 90 stupňů
posuň se o 1 kroků
otoč se o 90 stupňů
posuň se o 2 kroků
otoč se o 90 stupňů
posuň se o 2 kroků
  
```


C)

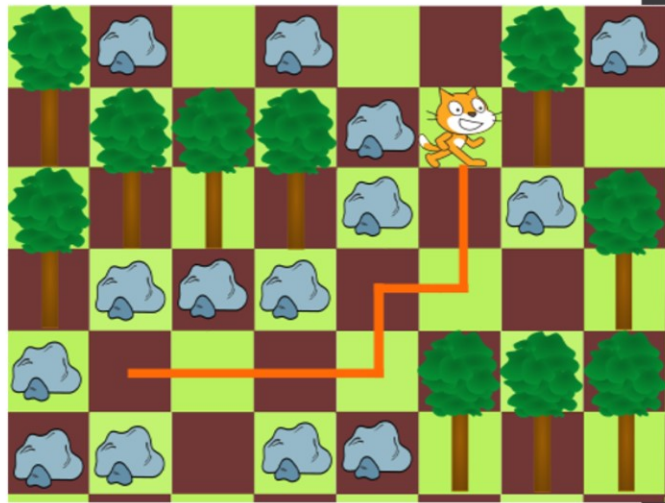
```

po kliknutí na [ ]
posuň se o 3 kroků
otoč se o 90 stupňů
posuň se o 1 kroků
otoč se o 90 stupňů
posuň se o 1 kroků
otoč se o 90 stupňů
posuň se o 2 kroků
  
```

ŘEŠENÍ

C)

```
po kliknutí na   
posuň se o 3 kroků  
otoč se o 90 stupňů  
posuň se o 1 kroků  
otoč se o 90 stupňů  
posuň se o 1 kroků  
otoč se o 90 stupňů  
posuň se o 2 kroků
```



Pomozte
jednorozci
najít duhu!



A)

```

    po kliknutí na
    posuň se o 3 kroků
    otoč se o 90 stupňů
    posuň se o 1 kroků
    otoč se o 90 stupňů
    opakuj 3 krát
    posuň se o 1 kroků
    otoč se o 90 stupňů
    posuň se o 2 kroků
    otoč se o 90 stupňů
    opakuj 3 krát
    posuň se o 1 kroků
  
```

B)

```

    po kliknutí na
    opakuj 2 krát
    posuň se o 1 kroků
    otoč se o 90 stupňů
    posuň se o 1 kroků
    otoč se o 90 stupňů
    opakuj 2 krát
    posuň se o 1 kroků
    otoč se o 90 stupňů
    opakuj 2 krát
    posuň se o 1 kroků
    otoč se o 90 stupňů
    opakuj 2 krát
    posuň se o 1 kroků
    otoč se o 90 stupňů
    opakuj dokud nenastane dotýká se barvy
    posuň se o 1 kroků
  
```

C)

```

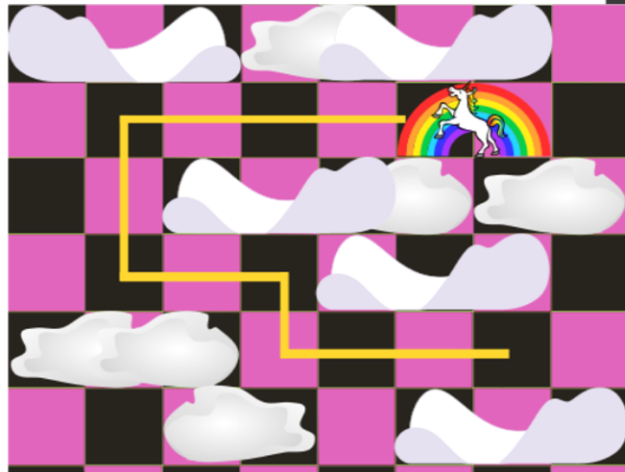
    po kliknutí na
    opakuj 3 krát
    posuň se o 1 kroků
    opakuj 2 krát
    otoč se o 90 stupňů
    posuň se o 1 kroků
    otoč se o 90 stupňů
    posuň se o 1 kroků
    otoč se o 90 stupňů
    opakuj 4 krát
    posuň se o 1 kroků
  
```

ŘEŠENÍ

B)

```

    po kliknutí na
    opakuj 3 krát
    posuň se o 1 kroků
    otoč se o 90 stupňů
    posuň se o 1 kroků
    otoč se o 90 stupňů
    opakuj 2 krát
    posuň se o 1 kroků
    otoč se o 90 stupňů
    opakuj 2 krát
    posuň se o 1 kroků
    otoč se o 90 stupňů
    opakuj dokud nenastane dotýká se barvy
    posuň se o 1 kroků
  
```



Zamyslete se a
poskládejte co nejvíce
fungujících algoritmů.
Nezapomeňte, že každý
algoritmus musí mít
začátek.