

Univerzita Karlova

Pedagogická fakulta

Katedra informačních technologií a technické výchovy

BAKALÁŘSKÁ PRÁCE

Možnosti využití aplikace Pencil Code ve výuce

Pencil Code in education

Karolína Gawłowska

Vedoucí práce: PhDr. Jiří Štípek Ph.D.

Studijní program: Specializace v pedagogice (B7507)

Studijní obor: B IT (7507R040)

2022

Prohlašuji, že jsem bakalářskou práci na téma **Možnosti využití aplikace Pencil Code ve výuce** vypracovala pod vedením vedoucího práce samostatně za použití v práci uvedených pramenů a literatury. Dále prohlašuji, že tato práce nebyla využita k získání jiného nebo stejného titulu.

Praha 10.07.2022

.....

podpis

Ráda bych touto cestou vyjádřila poděkování PhDr. Jiřímu Štípkovi Ph.D. za odborné vedení, cenné rady a věcné připomínky, které mi pomohly závěrečnou práci zkompletovat. Dále bych chtěla také poděkovat Ing. Martinu Kijonkovi Ph.D za odborný dohled a vstřícnost. Děkuji také PhDr. Marku Paulíkovi za veškerou podporu během mého studia.

ANOTACE

Tato bakalářská práce se snaží poukázat na možnosti využití aplikace Pencil Code ve výuce na ZŠ pro rozvoj algoritmického myšlení. Aplikace Pencil Code byla vyvinuta pro rozvoj algoritmického myšlení žáků a umožňuje široké možnosti pro výuku programování. Žáci mohou v aplikaci tvořit i složitější geometrické obrazce pomocí různých algoritmických konstrukcí.

Praktická část obsahuje ucelený soubor lekcí určených pro žáky v 5. – 6. třídě ZŠ. Vybrané úlohy byly ověřeny na ZŠ v 5. třídě. Pro následné využití lekcí jsem vytvořila webovou stránku v prostředí Pencil Code, která je volně dostupná pro učitele na adrese: <http://gawl.pencilcode.net/edit/index.html>.

KLÍČOVÁ SLOVA

Pencil Code, algoritmické myšlení, programování na základní škole

ANNOTATION

This bachelor thesis tries to show the possibilities of using Pencil Code in primary school education for the development of algorithmic thinking. The Pencil Code application has been developed to develop students' algorithmic thinking and allows a wide range of possibilities for teaching programming. Pupils can also create more complex geometric patterns in the app using various algorithmic constructions.

The practical part contains a comprehensive set of lessons designed for pupils in grades 5. – 6. of primary school. Selected problems have been tested in primary school in grade 5. For subsequent use of the lessons, I have created a web page in the Pencil Code environment, which is freely available for teachers at: <http://gawl.pencilcode.net/edit/index.html>.

KEYWORDS

Pencil Code, algorithmic thinking, programming at primary school

Obsah

1	Úvod	9
2	Cíl práce	10
2.1	Dílčí cíle práce:	10
3	Teoretická část	11
3.1	Algoritmické myšlení	11
3.2	Rámcově vzdělávací program a Školní vzdělávací program	14
	RVP ZV v souvislosti s algoritmickým myšlením	14
	Informatika	14
	Matematika a její aplikace	16
3.3	Možnosti využití Pencil Code na ZŠ	17
3.4	Pencil Code	17
	Prostředí	18
	Kategorie příkazů	19
	Výhody	20
	Nevýhody	21
4	Praktická část	22
4.1	Návrh úloh pro zjištění úrovně algoritmického myšlení	22
4.2	Lekce 1 – Základy	22
	Vstupní požadavky na žáky:	22
	Nové dovednosti:	22
	Obecná doporučení	23
	Seznámení s prostředím	23
	Čtverec	24
	Zaoblený čtverec	25

Kružnice.....	26
Trojúhelník	28
Dva tvary vzdálené od sebe	29
4.3 Cvičení k lekci 1	32
Tři čtverce.....	32
Kříž.....	32
Očička.....	33
Domeček.....	33
4.4 Lekce 2 – Základy.....	34
Vstupní požadavky na žáky:.....	34
Nové dovednosti:	34
Tečky různými barvami.....	34
Tečko-čtverec	35
Terč.....	37
Vlajka	39
Trojúhelník uvnitř kruhu	40
4.5 Cvičení k lekci 2	42
Duha z teček	42
Kruh ve čtverci s tečkami	42
Kapka s bambulí	43
Semafor.....	43
4.6 Doplnková cvičení (k lekcím 1 a 2).....	44
Smajlík.....	44
Lod'ka	44
Opičák.....	45

YinYang	45
BMW	46
4.7 Lekce 3.....	46
Vstupní požadavky na žáky:.....	46
Nové dovednosti:.....	46
Výklad k souřadnicím.....	46
Cvičení k souřadnicím.....	47
4.8 Doplnková cvičení (k lekcím 1, 2 a 3).....	48
Mimoň	48
Sněhulák	49
Pavouk	49
4.9 Lekce 4 - Cykly.....	50
Vstupní požadavky na žáky:.....	50
Nové dovednosti:.....	50
Čtverec pomocí cyklu.....	50
Tvary.....	51
Schody	52
Desetiúhelník.....	53
4.10 Cvičení k lekci 4.....	55
Trojúhelníky	55
Pila.....	56
Čtverec s tečkami.....	56
Audi	57
4.11 Doplnková cvičení (k lekci 4).....	57
Vločka.....	57

Hvězda	58
Hvězda2	58
4.12 Lekce 5 – Funkce	59
Vstupní požadavky na žáky:.....	59
Nové dovednosti:.....	59
Vlajky	59
4.13 Cvičení k lekci 5.....	63
Dopravní značky.....	63
4.14 Lekce 6 – Klávesy	63
Vstupní požadavky na žáky:.....	63
Nové dovednosti:.....	64
Výklad ke klávesám.....	64
4.15 Cvičení k lekci 6.....	65
Základní geometrické obrazce pomocí kláves.....	65
4.16 Lekce 7 – Složitější konstrukce Cykly 2.....	66
Vstupní požadavky na žáky:.....	66
Nové dovednosti:.....	66
Kytička z kružnic.....	66
Mašle	68
4.17 Doplnková cvičení (k lekci 7).....	69
Hodiny	70
AZ-kvíz.....	70
Davidova hvězda	71
5 Ověření	73
5.1 Ověření daných lekcí	73

Lekce 1	73
Lekce 2	75
Lekce 3	76
Lekce 4	77
5.2 Zhodnocení ověření lekcí.....	77
Soustředěnost.....	78
Zájem o činnost v prostředí	78
Vytrvalost	78
Emocionální stav	79
Samostatnost.....	79
Úspěšnost.....	79
Úpravy lekcí na základě ověřování	79
6 Závěr	81
7 Seznam použitých informačních zdrojů	83
8 Seznam obrázků.....	84
9 Seznam zkratk.....	88
10 Seznam příloh	89

1 Úvod

Tato bakalářská práce se snaží poukázat na možnosti využití aplikace Pencil Code ve výuce na základních školách pro rozvoj algoritmického myšlení.

V roce 2021 totiž proběhla revize RVP ZV – hlavního kurikulárního dokumentu. Cílem této revize bylo mimo jiné zavedení nové vzdělávací oblasti Informatika, jejíž nedílnou součástí je právě algoritmizace, resp. rozvoj algoritmického myšlení.

Aplikace Pencil Code byla vyvinuta pro rozvoj algoritmického myšlení žáků a umožňuje široké možnosti pro výuku programování. Aplikace nabízí tři výukové režimy, ve kterých mohou žáci tvořit. V mé bakalářské práci se zabývám režimem Draw, ve kterém mohou žáci vytvářet základní geometrické obrazce, které lze dále kombinovat a vytvářet z nich složitější tvary. Při vlastní tvorbě žáci využívají i základů geometrie a souřadný systém. Veškerá tvorba je možná v barevném provedení. Mezi nesporné výhody této aplikace patří její intuitivní ovládání, přehlednost a interaktivita. Po stisknutí tlačítka play mají žáci možnost ihned shlédnout výstup své tvorby.

Praktická část obsahuje ucelený soubor lekcí určených pro žáky v 5. – 6. třídě ZŠ. Úlohy jsou na sebe navazující, a to od úplných základů až po ty obtížnější. Žáci si postupně rozšiřují své dovednosti. Úlohy jsou rozděleny do jednotlivých lekcí, které obsahují jak teoretický výklad, tak i praktické procvičování. Jsou zde také doplňkové úlohy pro šikovnější žáky nebo na procvičení doma.

Vybrané úlohy z lekcí 1, 2, 3 a 4 byly ověřeny na základní škole v 5. třídě. Pro následné využití lekcí jsem vytvořila webovou stránku v prostředí Pencil Code, která je volně dostupná pro učitele. Webová stránka je zaměřena jak na lekce výkladové, tak i na lekce procvičovací. Stránka je dostupná na adrese: <http://gawl.pencilcode.net/edit/index.html>.

2 Cíl práce

Hlavním cílem bakalářské práce je analyzovat a ověřit možnosti využití aplikace Pencil Code ve vzdělávání pro rozvoj algoritmického myšlení u žáků. Tento hlavní cíl je dále rozdělen na cíle dílčí.

2.1 Dílčí cíle práce:

1. Analyzovat možnosti využití Pencil Code ve vzdělávání, zejména pro rozvoj algoritmického myšlení žáků základních škol.
2. Pro zvolenou věkovou skupinu připravit ucelený soubor aktivit orientovaných zejména na rozvoj algoritmického myšlení.
3. Ověřit navržený soubor aktivit v praxi.

3 Teoretická část

Vzhledem k zaměření mé bakalářské práce na téma využití aplikace Pencil Code ve výuce na základních školách zejména pro rozvoj algoritmického myšlení žáků se podrobněji seznámíme v této teoretické části s pojmy algoritmus a algoritmizace.

V podkapitole 3.2 jsou vysvětleny pojmy Rámcově vzdělávací program (RVP) a Školní vzdělávací program (ŠVP). V poslední podkapitole se zabýváme RVP ZV, aby bylo možné rozhodnout, do kterého ročníku připravené lekce zařadit. Vedle vzdělávací oblasti Informatika se zabýváme i vzdělávací oblastí Matematika a její aplikace, neboť Pencil Code pracuje s řadou matematických konceptů.

3.1 Algoritmické myšlení


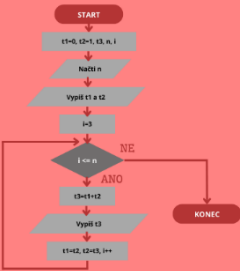
Pro porozumění pojmu algoritmického myšlení je potřeba si vysvětlit pojmy algoritmizace a algoritmus. Začneme tedy algoritmem, který můžeme definovat jako posloupnost příkazů, které řeší daný problém v konečném množství kroků.

„Algoritmus je jednoznačný srozumitelný přesně specifikovaný postup (předpis) vedoucí v konečném počtu kroků k řešení určité úlohy.“ (Havelková, 2008, str. 5)

Vzhledem k jednoduššímu pojetí předchozí definice algoritmus bych ráda uvedla úplnou definici z pohledu programátora.

„Algoritmus je přesně určená konečná posloupnost instrukcí, jejichž realizace nám umožní pro přípustné hodnoty vstupních dat nalézt po konečném počtu kroků správná výstupní data.“ (Wirth 1989, cit v Havelková 2017, str. 5)

Algoritmus můžeme zapisovat různými způsoby. První způsob zápisu je přirozený jazyk (návody nebo kuchyňské recepty). Druhým způsobem je jazyk strukturovaný. Strukturovaný jazyk je v podstatě stejný jako jazyk přirozený, ale platí zde určitá dohodnutá omezení (smíme používat silně omezené množství slovních výrazů). Třetí způsob je zápis v daném programovacím jazyce. Poslední způsob pro vyjádření algoritmu je způsob grafický (obrázky, diagramy nebo UML). (Štípek, 2020)

Zápis algoritmu: Fibonacciho posloupnost			
Slovní popis	Spirála	Vývojový diagram	Programovací jazyk
<p>Nekonečná posloupnost přirozených čísel, kdy každé číslo je součtem dvou předchozích</p> <p>1, 1, 2, 3, 5, 8, 13, 21, 34, 55 ...</p> <p>1+1=2 5+8=13 1+2=3 8+13=21 2+3=5 13+21=34 3+5=8 21+34=55</p>			<pre>#include <stdio.h> int main() { int i, n, t3; int t1 = 0, t2 = 1; printf("Kolik chces cisel: "); scanf("%d", &n); printf("Fibonacci: %d, %d, ", t1, t2); for (i = 3; i <= n; i++) { t3 = t1 + t2; printf("%d, ", t3); t1 = t2; t2 = t3; } return 0; }</pre>

Obrázek 1-Příklad zápisu algoritmu (zdroj autorka)

Ne každý návod či postup lze považovat za algoritmus. Algoritmus musí splňovat tyto požadavky: obecnost, konečnost, determinovanost, elementárnost a resultativnost.

Obecnost (hromadnost): Algoritmus by měl být použitelný pro celou třídu problému, tudíž neřeší jeden konkrétní problém (například 2+1).

Konečnost: Počet kroků algoritmu musí být konečný (nezacyklí se).

Determinovanost: Algoritmus by měl být postavený tak, že po každém jeho provedeném kroku musí být jednoznačné, jaký krok bude následovat (případně zda algoritmus skončil).

Elementárnost: Algoritmus je složený ze základních kroků, které jsou zcela pochopitelné pro vykonavatele.

Resultativnost: Algoritmus musí mít po skončení počtu kroků alespoň jeden výstup.

Algoritmy lze sestavit pomocí tří základních algoritmických konstrukcí – sekvencí příkazů, podmínek a cyklů. **Sekvence příkazů** je posloupnost kroků, které se postupně provedou. **Podmínky** rozdělí na základě vyhodnocení program do dvou větví. **Cykly** slouží k opakování příkazů. Opakování může být opatřené podmínkou buď na začátku, nebo na konci cyklu či opakování může být s pevným počtem. (Štípek, 2020)

S algoritmy se setkáváme v každodenním životě. Většinou je pojem spojován spíše s informačními technologiemi, ale jejich užívání v běžném životě je zcela běžné. Jsou to návody na sestavení nábytku, kuchyňský recept nebo další manuály popisující určité postupy

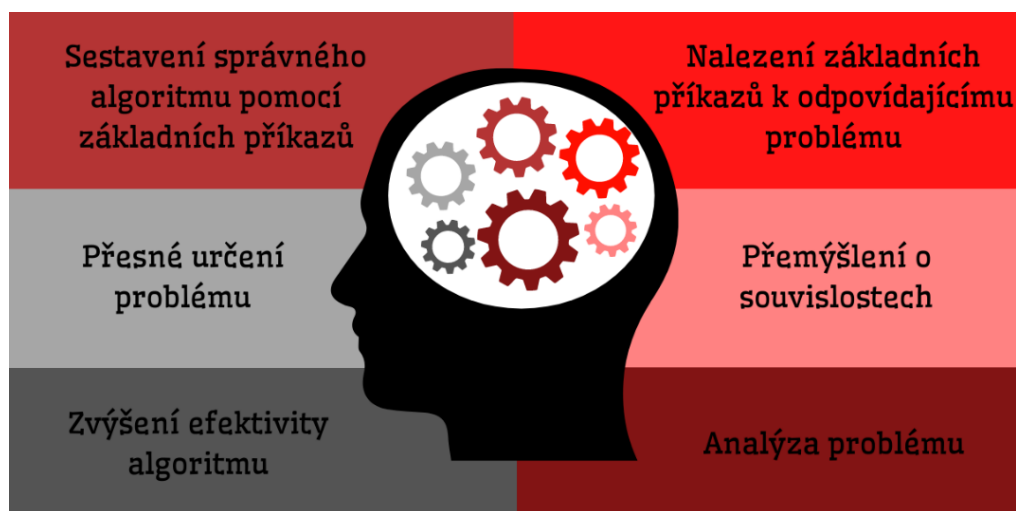
řešení různých problémů. Tyto algoritmy musejí obsahovat zcela srozumitelné a jednoznačné kroky pochopitelné pro běžného člověka.

Algoritmizace je způsob či návrh, jak sestavit algoritmus, který splňuje všechny výše uvedené požadavky.

„Algoritmizace je přístup k vytváření programu. Zabývá se formulací postupů řešení daného problému. Výsledkem algoritmizace je algoritmus, což je posloupnost příkazů popisující řešení daného problému.“ (Dohnal, 2008, str. 10)

Dále bych v této části práce chtěla vymezit pojem algoritmické myšlení. Existuje celá řada definic pojmu algoritmické myšlení. Algoritmické myšlení je definováno jako soubor schopností, které jsou spojeny s konstruováním algoritmů a porozuměním jim. Mezi tyto schopnosti patří:

- přesné určení problému,
- analýza problému,
- nalezení základních příkazů k odpovídajícímu problému,
- sestavení správného algoritmu pomocí základních příkazů,
- přemýšlení o souvislostech,
- zvýšení efektivity algoritmu. (Futschek, 2006)



Obrázek 2- Algoritmické myšlení (zdroj autorka)

Algoritmické myšlení s výjimkou výše uvedených schopností vyžaduje i další dovednosti. Konkrétně abstraktní myšlení, logické myšlení, strukturované myšlení, tvořivost a schopnost řešit dané problémy. Toto nás vede k tomu, že není zcela jednoduché si dovednost algoritmického myšlení osvojit, nýbrž mu musíme skutečně porozumět. (Futschek, 2006)

3.2 Rámcově vzdělávací program a Školní vzdělávací program

Rámcově vzdělávací program (RVP) je kurikulární dokument, který definuje ve školství v České republice nejvyšší úroveň vzdělávání. Dříve spolu s Národním programem pro rozvoj vzdělávání, který přestal platit k roku 2020. Na základě tohoto rozhodnutí došlo ke změně kurikulárních dokumentů, které jsou vytvářeny na dvou úrovních (**státní a školní**). (RVP ZV, 2021, str. 5)

Státní úroveň představuje Rámcově vzdělávací programy, které vymezují rámce vzdělávání jednotlivých etap (předškolní, základní a střední vzdělávání) a slouží jako předloha pro tvorbu školních vzdělávacích programů. **Školní úroveň** představuje Školní vzdělávací programy, podle nichž se uskutečňuje výuka na jednotlivých školách. Oba kurikulární dokumenty jak RVP, tak ŠVP jsou přístupné pro pedagogickou i nepedagogickou veřejnost. (RVP ZV, 2021, str. 5)

RVP ZV v souvislosti s algoritmickým myšlením

Informatika

V oblasti Informatiky došlo k výrazným změnám oproti roku 2017. Lze si povšimnout už i změněného názvu vzdělávací oblasti Informační a komunikační technologie na zmiňovanou vzdělávací oblast Informatika.

„Vzdělávací oblast Informatika se zaměřuje především na rozvoj informatického myšlení a na porozumění základním principům digitálních technologií. Je založena na aktivních činnostech, při kterých žáci využívají informatické postupy a pojmy. Poskytuje prostředky a metody ke zkoumání řešitelnosti problémů i hledání a nalézání jejich optimálních řešení, ke zpracování dat a jejich interpretaci a na základě řešení praktických úkolů i poznatky a zkušenost, kdy je lepší práci přenechat stroji, respektive počítači. Pochopení, jak digitální technologie fungují, přispívá jednak k porozumění zákonitostem digitálního světa, jednak k jejich efektivnímu, bezpečnému a etickému užívání.“ (RVP ZV, 2021, str. 38)

RVP ZV konkrétně zmiňuje ve vzdělávací oblasti Informatiky algoritmizaci už na prvním stupni základního vzdělávání: „Postupně si žáci rozvíjejí schopnost popsat problém, analyzovat ho a hledat jeho řešení. Ve vhodném programovacím prostředí si ověřují algoritmické postupy.“ (RVP ZV, 2021, str. 38)

Očekávané výstupy žáka prvního stupně v oblasti Informatika pro výuku Algoritmizace a programování:

- „sestavuje a testuje symbolické zápisy postupů“
- „popíše jednoduchý problém, navrhne a popíše jednotlivé kroky jeho řešení“
- „v blokově orientovaném programovacím jazyce sestaví program; rozpozná opakující se vzory, používá opakování a připravené podprogramy“
- „ověří správnost jím navrženého postupu či programu, najde a opraví v něm případnou chybu“ (RVP ZV, 2021, str. 39)

Očekávané výstupy žáka druhého stupně v oblasti Informatika pro výuku Algoritmizace a programování:

- „po přečtení jednotlivých kroků algoritmu nebo programu vysvětlí celý postup; určí problém, který je daným algoritmem řešen“
- „rozdělí problém na jednotlivě řešitelné části a navrhne a popíše kroky k jejich řešení“
- „vybere z více možností vhodný algoritmus pro řešený problém a svůj výběr zdůvodní; upraví daný algoritmus pro jiné problémy, navrhne různé algoritmy pro řešení problému“
- „v blokově orientovaném programovacím jazyce vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za ně; program vyzkouší a opraví v něm případné chyby; používá opakování, větvení programu, proměnné“
- „ověří správnost postupu, najde a opraví v něm případnou chybu“
(RVP ZV, 2021, str. 41)

Z výše uvedeného dokumentu RVP ZV (2021) vyplývá, že žáci začínají už v pátém ročníku vyvíjet funkční řešení problémů a vylepšují své postupy. Zvažují a ověřují dopady navrhovaných řešení.

Matematika a její aplikace

RVP ZV zmiňuje pojem algoritmické myšlení také ve vzdělávací oblasti Matematika a její aplikace. Zde však od roku 2017 nedošlo k žádným změnám.

„Vzdělávání klade důraz na důkladné porozumění základním myšlenkovým postupům a pojmům matematiky a jejich vzájemným vztahům. Žáci si postupně osvojují některé pojmy, algoritmy, terminologii, symboliku a způsoby jejich užití. Vzdělávací obsah vzdělávacího oboru Matematika a její aplikace je rozdělen na čtyři tematické okruhy. V tematickém okruhu Čísla a početní operace na prvním stupni, na který na druhém stupni navazuje a dále ho prohlubuje tematický okruh Číslo a proměnná, si žáci osvojují aritmetické operace v jejich třech složkách: dovednost provádět operaci, algoritmické porozumění (proč je operace prováděna předloženým postupem) a významové porozumění (umět operaci propojit s reálnou situací). Učí se získávat číselné údaje měřením, odhadováním, výpočtem a zaokrouhlováním. Seznamují se s pojmem proměnná a s její rolí při matematizaci reálných situací.“ (RVP ZV, 2021, str. 30)

Cílové zaměření vzdělávací oblasti Matematika a její aplikace RVP ZV zmiňuje pojem algoritmizace jako:

- *„vytváření zásoby matematických nástrojů (početních operací, algoritmů, metod řešení úloh) a k efektivnímu využívání osvojeného matematického aparátu“*
- *„rozvíjení paměti žáků prostřednictvím numerických výpočtů a osvojování si nezbytných matematických vzorců a algoritmů“ (RVP ZV, 2021, str. 30)*

Očekávaný výstup pro druhé období v oblasti Matematika a její aplikace žák *„řeší jednoduché praktické slovní úlohy a problémy, jejichž řešení je do značné míry nezávislé na obvyklých postupech a algoritmech školské matematiky“ (RVP ZV, 2021, str. 34)*

Hlavním cílem použití algoritmizace ve výuce na nižších stupních je celkový rozvoj tvořivosti. (Pitner, 2000, str. 3)

Tvořivost není samostatná vzdělávací oblast, která by se vyučovala na základních školách jako samostatný předmět, ale jako taková prostupuje celým RVP.

„Tvořivost je hybnou silou rozvoje lidstva.“ (Kijonka, 2022)

3.3 Možnosti využití Pencil Code na ZŠ

Na základě analýzy algoritmického myšlení RVP ZV ve vzdělávací oblasti Matematika a její aplikace a ve vzdělávací oblasti Informatika jsou pro práci v aplikaci Pencil Code nutné tyto výstupy žáka:

Očekávaný výstup žáka v prvním období učiva Geometrie v rovině a prostoru (5. třída):

- *„rozezná, pojmenuje, vymodeluje a popíše základní rovinné útvary a jednoduchá tělesa; nachází v realitě jejich reprezentaci“ (RVP ZV, 2021, str. 33)*

Očekávané výstupy žáka ve druhém období učiva Geometrie v rovině a prostoru (5. třída):

- *„narýsuje a znázorní základní rovinné útvary (čtverec, obdélník, trojúhelník a kružnici); užívá jednoduché konstrukce“*
- *„určí obsah obrazce pomocí čtvercové sítě a užívá základní jednotky obsahu“ (RVP ZV, 2021, str. 33)*

Ve vzdělávací oblasti Informatika žák nepotřebuje disponovat předešlými znalostmi.

Žáci nepotřebují žádné nebo jen minimální znalosti programování. Postačí pouhá znalost ovládání počítače a vyhledávání na internetu. RVP ZV předpokládá, že žáci již vyrostli do prostředí, kde se většinou běžně komunikuje pomocí mobilních telefonů či dalšími zařízeními (notebook, tablet). Předpokládám tedy, že žáci v 5. třídě budou mít postačující úroveň digitální kompetence k užívání aplikace Pencil Code.

Na základě analýzy RVP ZV a ŠVP ZŠ Nad Přehradou (viz příloha A), kde probíhalo ověřování mých připravených lekcí v aplikaci Pencil Code, navrhuji tedy zařazení těchto lekcí v aplikaci Pencil Code do výuky Informatiky (Algoritmizace a programování) 5. třídy ZŠ.

3.4 Pencil Code

Pencil Code je online vývojové prostředí, které nabízí tři výukové režimy (Draw, Jam, Imagine), pomocí kterých můžeme zápisem kódu (CoffeeScript) malovat, tvořit hudbu či vytvořit příběh či hru. Pencil Code vzniknul v roce 2001 za spolupráce dvou kolegů – Axelem Brandenburgem a Wolfgangem Doblerem. Dle tvůrců je určen k výuce informatiky

a algoritmického myšlení, tím že programování má učinit tak jednoduchým a univerzálním jako je používání prostého pera.

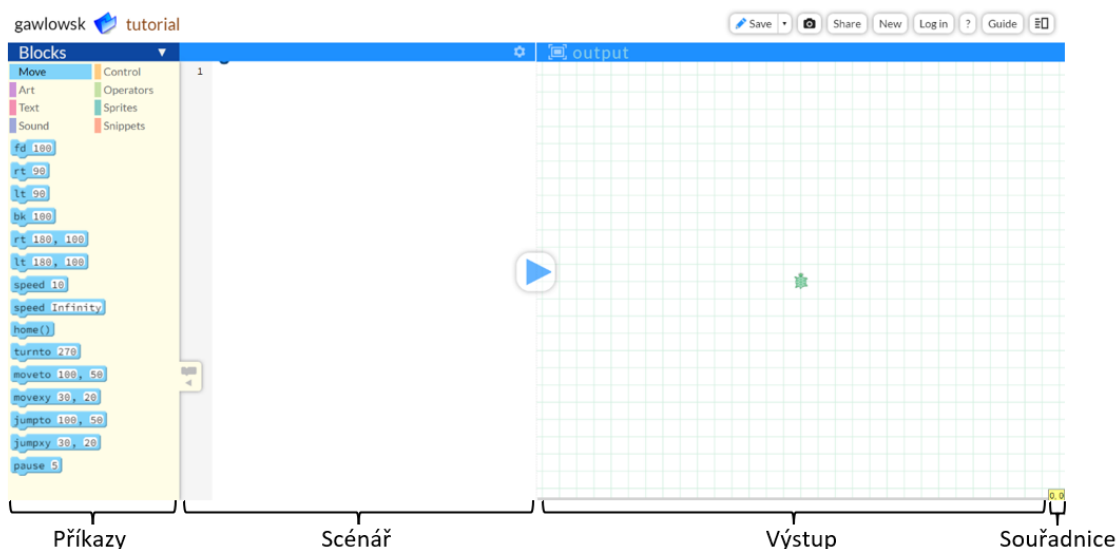
Ve všech svých režimech používá vizuální jazyk v podobě do sebe zapadajících bloků. Je zde prostor také pro tvorbu geometrických konstrukcí, vytváření matematických funkcí, grafů či tvorbu webové stránky (HTML, Javascript a CSS).

Má bakalářská práce tedy využívá možnosti online aplikace dostupné na adrese www.pencilcode.net a mnou vytvořené úlohy jsou v režimu Draw.

Prostředí

Prostředí je přehledné, uspořádané a uživatel se v něm snadno a rychle zorientuje. I když se jedná spíše o lineární prostředí, tak i přesto nabízí nepřehledné množství možností, kterých může kreativní uživatel využít ke své tvorbě. Mohou tak vzniknout i rozsáhlé projekty.

Jak je patrné z Obrázek 3, prostředí je rozděleno na tři části. Vlevo se nachází příkazy v různých kategoriích v režimu **blocks**. Uprostřed je prostor pro tvorbu našeho scénáře, kam se přetahují bloky s příkazy. Lze přepnout na režim **text code**, kde se píše příkazy podle správné syntaxe. Mezi kategoriemi s příkazy lze také přepínat. Vpravo je výstup, kde se nachází prostředník pro vykonávání příkazů – a to želvička, která koná na základě scénáře, tedy algoritmu. Pro popis, resp. určení pozice na výstupu je využita kartézská soustava souřadnic s počátkem v geometrickém středu výstupu. Želvička vždy začíná uprostřed výstupu, tedy na souřadnicích [0;0]. Pro snazší orientaci, je na pozadí výstupu čtverečková síť, a navíc se v pravém dolním rohu zobrazují aktuální souřadnice kurzoru myši.



Obrázek 3-Prostředí Pencil Code

V Pencil Code se také nachází horní lišta, která zpřístupňuje řadu funkcí v prostředí. V pravém horním rohu jsou tlačítka pro ukládání projektu, screenshot, sdílení, vytvoření nového projektu, přihlášení uživatele, odhlášení uživatele a nápovědu. Vlevo vidíme jméno založeného účtu, hlavní adresář a projekt, ve kterém se nacházíme (viz Obrázek 4).



Obrázek 4-Hlavní lišta Pencil Code

Kategorie příkazů

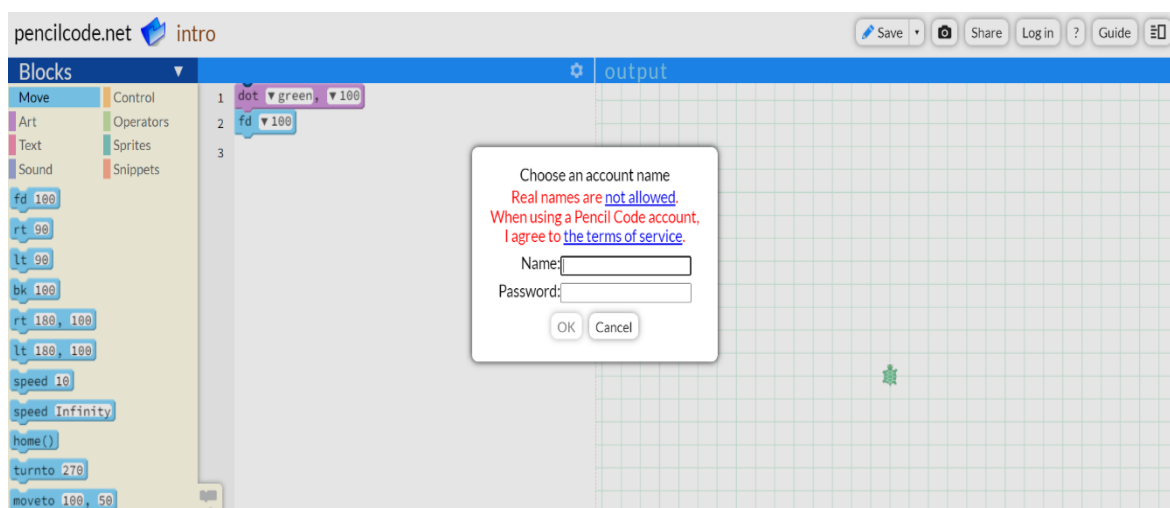
Pro tvorbu scénářů uživatel může využít příkazy z různých kategorií. Všechny kategorie, tudíž i příkazy v nich vnořené, jsou rozdělené barevně podle činností. Přetahování příkazů do scénáře funguje na principu „drag and drop“. Skládání kódu tak připomíná skládání puzzle.

- **Move** (pohyb, značen modře) obsahuje příkazy pro pohyb jako posun dopředu a dozadu, otáčení o úhel, rychlost nebo přesouvání se na určitou pozici v soustavě (osa X a Y).
- **Art** (vzhled, značen fialově) obsahuje příkazy pro definování vzhledu, vytvoření tečky a čtverečku, nastavení jejich velikosti a barvy, nastavení tloušťky obrysu a barvy pera. Postavičku želvičky lze zvětšovat, měnit, skrýt nebo pomocí ní ji aktivovat a deaktivovat pero.

- **Text** (text, značen růžově) obsahuje příkazy spojené s textem.
- **Sound** (zvuk, značen modro-fialově) obsahuje příkazy spojené se zvukem.
- **Control** (ovládání, značeno oranžově) obsahují bloky pro tvorbu cyklů a podmínek. Dále jsou zde příkazy pro zachycení události (stisknutí klávesy, tlačítka či klik myši).
- **Operators** (operátory, značeny zeleně) obsahují matematické a logické operátory, proměnné a bloky pro tvorbu funkcí.
- **Sprites** (značen tmavě zeleně) obsahuje příkazy pro vytvoření nových objektů.
- **Snippets** (útržky, značeny tomatově) v této kategorii najdeme příklady a části hotového kódu.

Výhody

Pencil Code je zdarma a online. Žák nemusí nic stahovat a instalovat, ale rovnou tvořit. Stačí přejít na web <http://pencilcode.net/> a zaregistrovat se. Registrace je velmi snadná a intuitivní. Žák nemusí disponovat emailovou adresou, stačí pouze vyplnit jméno či přezdívku a heslo. Po vytvoření účtu si žák může své projekty archivovat a doma upravovat či dokončovat úlohy.



Obrázek 5-Screenshot ukázky registrace

Jako pozitivum lze hodnotit animovaný pohyb želvičky, díky kterému můžeme sledovat její posuny a otáčení. Můžeme také nastavovat rychlost pohybu želvy a její poloměr otáčení.

Díky barevně odděleným kategoriím se žáci naučí lehce rozlišovat dané příkazy. Kód se skládá pomocí bloků, které do sebe zapadají jako puzzle. Systém skládání bloků do sebe má výhodu v tom, že žák nemůže udělat chybu v syntaxi.

Po přetažení kódu do svého scénáře a stisknutím tlačítka play se spustí scénář. Žák ihned uvidí, co naprogramoval. Žák nemusí složitě hledat chyby ve svém kódu, protože vše má na jednom místě. Pro spuštění algoritmu není potřeba nic stahovat nebo spouštět exe soubory.

Nevýhody

Pencil Code nabízí jen jednojazyčnou verzi, a to anglický jazyk. Pokud žák nemá doma internetové připojení, tak zde není off-line verze pro stažení do počítače. Bohužel také neexistuje mobilní aplikace, aby žák mohl tvořit na tabletu či mobilním zařízení.

U jiných prostředí (např. Scratch) po kliknutí na příkaz v bance provede postava odpovídající akci, ale v Pencil Code to tak není. Příkazy (v režimu **blocks**) nelze jednoduše testovat, pouze je přetahovat na místo ve scénáři. Při spuštění programu není vidět, který z příkazů je právě prováděný. Příkaz ve scénáři totiž není interaktivně zvýrazněný a není tedy vidět, který je aktuálně želvou prováděn.

Barvy, jakožto důležitý aspekt, se určují pouze názvem nebo pomocí modelu RGB. V základním režimu zde máme jednu postavu a to želvičku, avšak želvička umí vzít na sebe jinou podobu (příkaz **wear**).

4 Praktická část

Cílem praktické části mé bakalářské práce bylo vytvořit ucelený soubor aktivit v aplikaci Pencil Code a jejich následné ověření na základní škole. Následně na základě ověření doporučit tento ucelený soubor aktivit k zařazení do výuky konkrétní vzdělávací oblasti.

4.1 Návrh úloh pro zjištění úrovně algoritmického myšlení

Vytvořené lekce jsou určeny pro žáky 5. a 6. tříd základní školy. Účelem těchto lekcí je rozvoj schopností žáků využívat algoritmické myšlení k řešení určitých zadání či úloh. Pro tento účel byly navrženy lekce, které obsahují úlohy zaměřené na algoritmizaci, matematiku, logiku a programování.

4.2 Lekce 1 – Základy

Vstupní požadavky na žáky:

- Žáci se pomocí první lekce seznamují s prostředím Pencil Code. Předchozí zkušenost není nutností.

Nové dovednosti:

- Naučí se zakládat projekty a složky ve svém adresáři, všechno si správně pojmenovat a ukládat.
- Naučí se přecházet mezi hlavním adresářem a stávajícím projektem.
- Důležitým aspektem je práce s bankou příkazů, přetahování příkazů, vnímání souvislostí mezi svým vytvořeným scénářem a následným výstupem.
- Osvojení si příkazů z kategorie **move**: jít rovně (**fd**), zatočení doprava (**rt**), zatočení doleva (**lt**), zatočení o úhel (**rt/lt**; poloměr otáčení **r**).
- Osvojení si příkazů z kategorie **art**: nadefinovat barvu a tloušťku obrysu pera (**pen**), vyplnění obrazce barvou (**fill**), zapnutí pera (**pen up**) a odebrání pera (**pen down**).
- Seznámení se s aritmetickými operátory a jejich následné užití v příkazech.

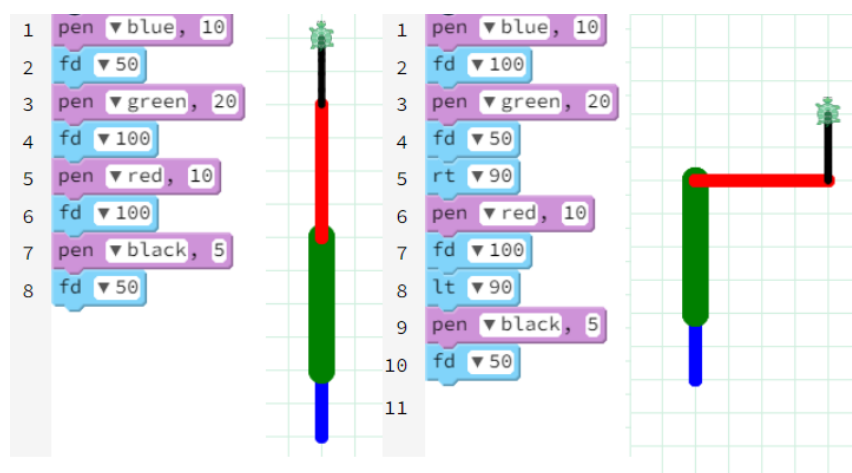
Obecná doporučení

Pro přehlednější práci v lekcích si vždy založíme nový projekt a řádně si ho pojmenujeme. Hotový program lze uložit, buď pomocí tlačítka **Save** nebo klávesovou zkratkou **Ctrl + S**.

Seznámení s prostředím

Při prvním spuštění prostředí si ukážeme příkaz pro pohyb želvy vpřed (**fd**), který si přetáhneme do našeho scénáře a spustíme. Ve výstupu uvidíme želvičku popojít o hodnotu 100. Zkusíme si přetáhnout další příkaz **fd** a následně mu změnit hodnotu. Opět uvidíme želvičku jít dopředu. Nyní budeme chtít, aby nám želvička začala zanechávat stopu. Použijeme proto příkaz pro kreslení perem (**pen**) z kategorie **art** a dáme ho před naše dva příkazy **fd**. Uvidíme, že nám už želvička kreslí předdefinovanou fialovou barvou a tloušťkou. Vyzkoušíme si změnit barvu a tloušťku.

Na začátek si žáci udělají několik různobarevných čárek různé tloušťky obrysu pera. Po prvotním seznámením s příkazy **fd** a **pen** si ještě ukážeme, jak zatáčet pomocí příkazu **rt/lt**. Přetáhneme proto příkazy zatáčení z kategorie **move** a vložíme je kamkoliv do našeho scénáře (viz Obrázek 6).



Obrázek 6-Odzkoušení příkazů pro pohyb a zatáčení želvy a nastavení pera

Tento projekt si uložíme a přejdeme do hlavního adresáře, kde si vytvoříme další projekt s názvem čtverec.

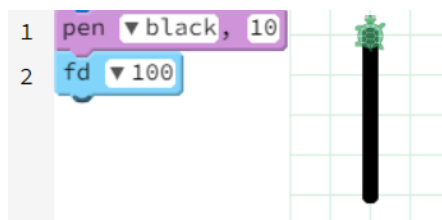
Čtverec

Čtverec je v geometrii brán jako pravidelný čtyřúhelník čili se jedná o rovinný útvar se čtyřmi shodnými stranami. Jeho všechny vnitřní úhly jsou taktéž shodné (mají 90°).



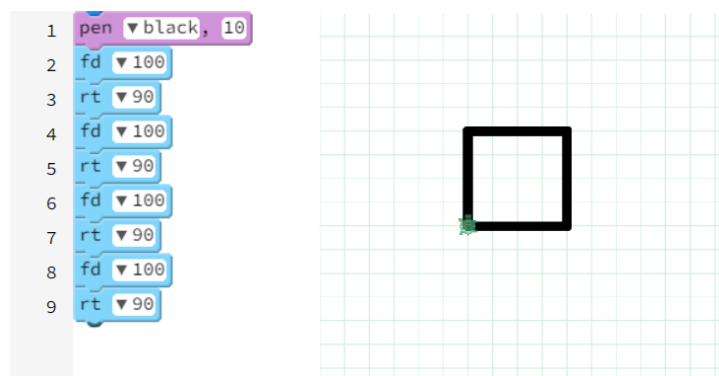
Obrázek 7-Čtverec

Do našeho scénáře si přetáhneme příkaz pro pohyb želvy (**fd**). Zvolíme si délku strany dle našich preferencí (nejlépe v rozmezí 50-200 bodů) a spustíme program. Ve výstupu uvidíme pohyb želvičky směrem nahoru. Načež zjistíme, že nám želvička nezanechává žádnou stopu, proto využijeme příkaz **pen** z kategorie **art**. Strana čtverce bude černou barvou a tloušťka obrysu bude 10 bodů. Ve scénáři máme tedy dva příkazy (**fd** a **pen**), které nám ve výstupu dají jednu stranu čtverce (viz Obrázek 8).



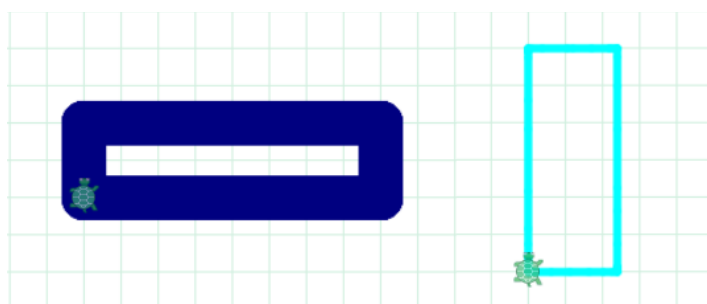
Obrázek 8-Nastavení pera a první strana čtverce

Dále využijeme příkaz zatočení, který se jmenuje **rt** (zatoč doprava) nebo **lt** (zatoč doleva). Ponecháme hodnotu 90, jako 90° . Spustíme program a uvidíme, že je želvička připravena k nakreslení další strany. Další stranu nakreslíme opět příkazem **fd**, poté přidáme totožné zatočení jako v předešlém kroku. Takto to zopakujeme ještě dvakrát.



Obrázek 9-Hotové řešení

Mini úkol: Pro vyzkoušení si žáci mohou každou stranu čtverce udělat jinou barvou nebo tloušťkou pera. Žáci si také mohou zkusit předělat výsledný čtverec na obdélník (viz Obrázek 10).



Obrázek 10-Mini úkol ke čtverci

Zaoblený čtverec

Dalším tvarem bude zaoblený čtverec. Můžeme použít náš již vytvořený čtverec nebo si založíme nový projekt.

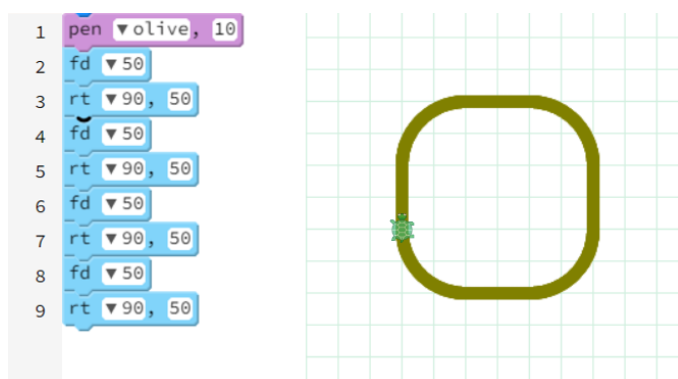


Obrázek 11-Zaoblený čtverec

Zaoblený čtverec se bude lišit od čtverce v příkazu zatočení. Odlišnost bude v tom, že místo prostého zatočení doprava nebo doleva (**rt/lt**), použijeme existující (předem definovaný)

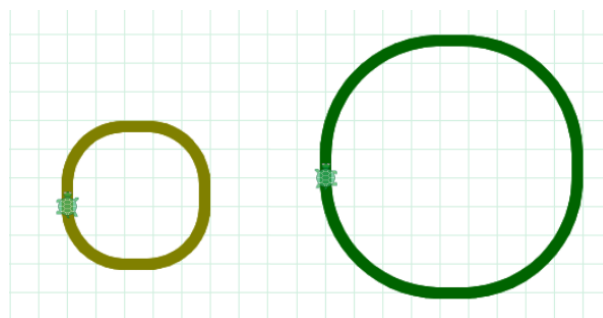
příkaz v kterém je kromě úhlu ještě poloměr otočení (**rt/lt**; poloměr otáčení **r**). Náš zaoblený čtverec bude nakreslený barvou **olive**.

Po nakreslení zaobleného čtverce příkazem **pen** z kategorie **art** a barvou **olive** odebereme příkazy **rt** a místo nich vložíme na stejné místo (tedy za každou stranu čtverce) příkaz zatočení doprava s úhlem 90 a poloměrem zaoblění 50 bodů. U našeho zaobleného čtverce ponecháme délku strany (50 bodů).



Obrázek 12-Hotové řešení

Mini úkol: Žáci si mohou vyzkoušet měnit parametry strany čtverce i délku poloměru u zaoblění. Můžeme žáky instruovat, aby zkusili postupně zkracovat stranu čtverce (**fd**) a zvětšovat poloměr zaoblění. Pomalu v tom mohou spatřit, že se čtverec, pomocí zaoblění vrcholů, může měnit na kružnici (viz Obrázek 13).



Obrázek 13-Mini úkol k zaoblenému čtverci

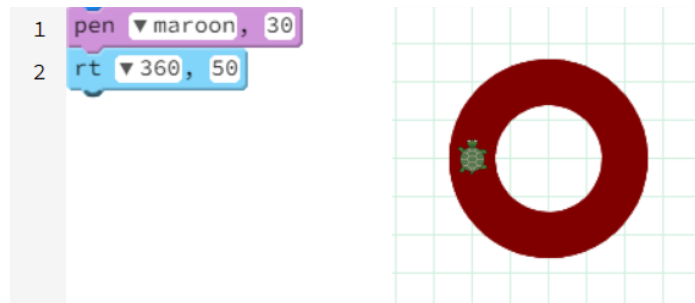
Kružnice

Kružnice je geometrický útvar tvořený množinou všech bodů ve stejné vzdálenosti od pevně daného bodu, který se nazývá střed.



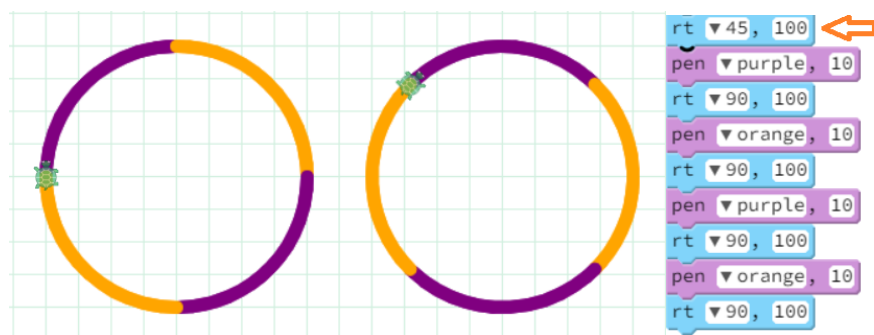
Obrázek 14-Kružnice

Než začneme přemýšlet, jak vytvořit kružnici, tak si vezmeme jako první příkaz **pen** z kategorie **art**. Vybereme si barvu a tloušťku pera. O kružnici víme, že má 360° , tudíž použijeme příkaz zatočení **rt** nebo **lt**. Spustíme program a uvidíme na výstupu, že se nám želvička jenom otočila kolem své osy. Jako u předchozí úlohy se zaoblením vrcholů u čtverce. Budeme potřebovat příkaz zatočení o úhel **rt/lt** s poloměrem otáčení **r**. Vypůjčíme si tedy příkaz z kategorie **move**, třeba zatočení doprava s předdefinovanými 180° a poloměrem 100 bodů. Na výstupu zpozorujeme, že se želvička otočila doprava a udělala takzvaně půlkružnici. Můžeme proto použít úplně totožný příkaz a zařadit ho pod nebo si lze uvědomit, že nám stačí přepsat číslo 180° na 360° .



Obrázek 15-Hotové řešení

Mini úkol: Pro procvičení si žáci mohou zkusit měnit barvu pera, tloušťku obrysu pera nebo směr točení kružnice. Žáci vytvoří kružnici, která bude rozdělena na čtyři části, kde každá část bude jinou barvou (viz Obrázek 16).



Obrázek 16-Mini úkol ke kružnici

Nápověda: Druhá kružnice má příkaz navíc. Jedná se o příkaz otočení o 45 stupňů stejného poloměru r .

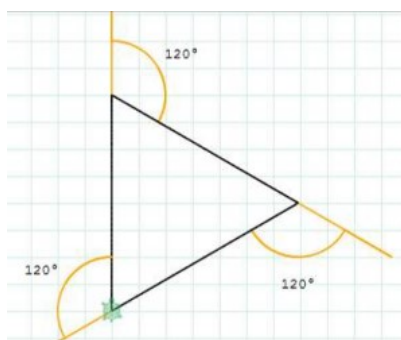
Trojúhelník

V geometrii je trojúhelník určený třemi body neležícími na jedné přímce. Součet jeho vnitřních úhlů nám dává 180° . Nejjednodušší pro nás bude si ukázat rovnostranný trojúhelník. Rovnostranný trojúhelník má všechny strany stejně dlouhé a jeho vnitřní úhly mají stejnou hodnotu (60 stupňů).



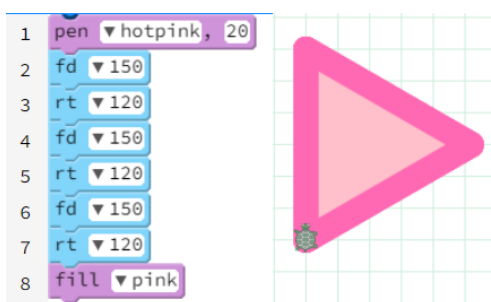
Obrázek 17-Rovnostranný trojúhelník

Začneme příkazem pro kreslení (**pen**), kterému nadefinujeme barvu pera **hotpink** a tloušťku obrysu pera 20 bodů. Na vytvoření strany použijeme z kategorie **move** příkaz **fd** s hodnotou 150. Nakreslíme tím první stranu, poté se budeme muset potočit příkazem **rt**. Želvička nám kouká směrem nahoru. Pro vytvoření vnitřního úhlu rovnostranného trojúhelníka (tedy 60 stupňů), bude potřeba želvičku vytočit o úhel vedlejší (180-60).



Obrázek 18-Obrázek k úhlům trojúhelníka

Tudíž ji vytočíme o 120 stupňů doprava příkazem **rt120**. Pro nakreslení dalších dvou stejných stran opět použijeme příkazy pro pohyb želvičky (**fd**) a otočení o úhel 120 stupňů (**rt 120**). Vybarvení vytvořeného obrazce dosáhneme příkazem **fill**, který je v kategorii **art**. Má parametr, který obsahuje jenom název barvy v našem případě **pink**.



Obrázek 19-Hotové řešení

Na vyzkoušení si děti nakreslí trojúhelník se základnou, která bude rovnoběžná s horizontálou. Lze udělat tak, že si želvičku nejdřív vytočí doleva o 90 stupňů a budou pokračovat stejným způsobem jako výše. Druhým způsobem může být vložení příkazu vytočení doprava o 30 stupňů před skupinu příkazů pro tvorbu trojúhelníka.

Dva tvary vzdálené od sebe

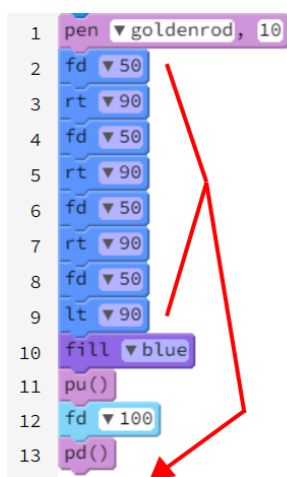
Na dvou čtvercích, které jsou od sebe vzdálené o nějaký úsek, si ukážeme nové příkazy. Příkazy pro odebrání a navrácení pera.



Obrázek 20-Dva čtverce pod sebou

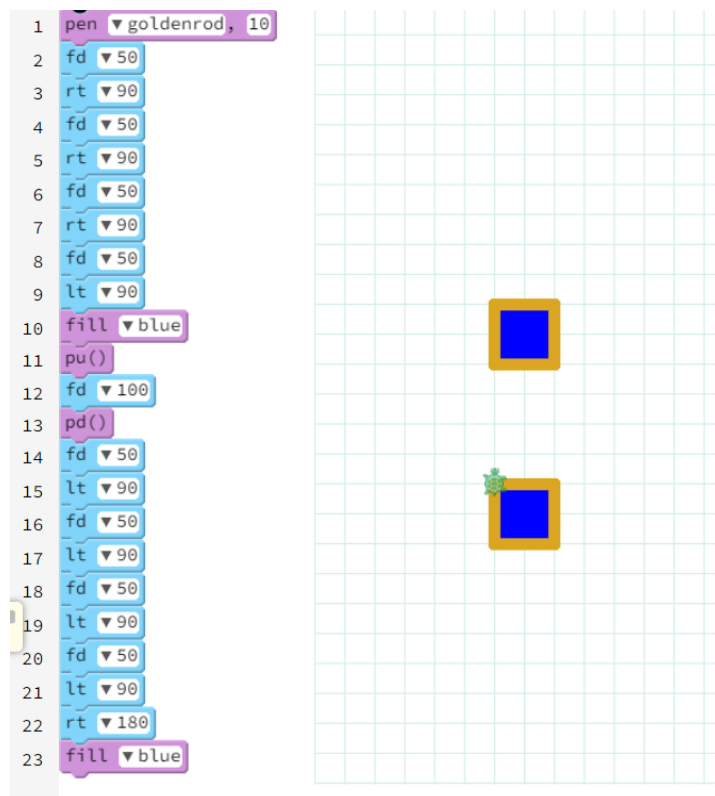
Vytvoříme si čtverec (viz první úloha-tvorba čtverce). Vybarvíme si čtverec příkazem **fill** z kategorie **art**. Dalším úkolem pak bude nakreslit další stejný čtverec pod našim vytvořeným. Toho docílíme tak, že si natočíme želvičku příkazem otočení a to o 90 nebo 180 stupňů podle toho, kterým směrem je želvička natočena po nakreslení poslední strany čtverce.

Máme ji tedy otočenou dolů. Aby se nám želvička posunula o kus dolů, už umíme. Pro vytvoření mezery mezi čtverci, musíme želvičce odebrat pero příkazem **pu** (pen up), který najdeme v kategorii **art**. Poté želvičku posuneme příkazem **fd** třeba o 100 bodů, tím uděláme mezeru. Zde přichází vhodná chvíle na kopírování příkazů. Veškeré příkazy, které souvisí s tvorbou čtverce si označíme přejetím myši a držením levého tlačítka myši. Příkaz nám zmodrá, což znamená, že ho máme označený (viz Obrázek 21).



Obrázek 21-Kopírování příkazů

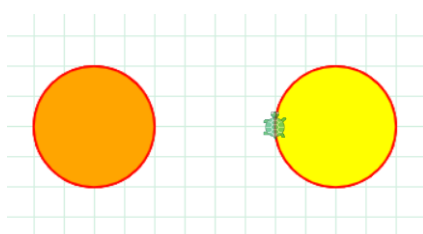
Označený příkaz zkopírujeme klávesovou zkratkou **Ctrl + C** a vložíme za příkaz **pd** klávesovou zkratkou **Ctrl + V**.



Obrázek 22-Hotové řešení

Mini úkol: Na vyzkoušení místo čtverců, můžeme vytvořit dva kruhy vedle sebe. Oba kruhy budou odlišně barevné, kde použijeme příkaz **fill** a budou od sebe vzdálené o nějaký kus.

Nápověda: Aby želvička nezanechala stopu při kreslení druhého kruhu, využijeme příkazy odebrání pera (**pu**), želvička se posune a opět navrátíme pero (**pd**) pro vytvoření dalšího kruhu. (viz Obrázek 23).



Obrázek 23-Mini úkol ke dvěma čtvercům

Alternativou k mini úkolu mohou být dva trojúhelníky. Děti si mohou zvolit, kterou úlohu udělají. Jedná se o dva trojúhelníky vzdálené od sebe. Oba vrcholy trojúhelníků směřují k sobě. (viz Obrázek 24)



Obrázek 24-Druhý mini úkol ke dvěma čtvercům

4.3 Cvičení k lekcí 1

Cvičení si vyučující vybere sám podle obtížnosti a hodinové dotace. Jednotlivá cvičení prověří zvládnutí této lekce. Vyučující může vytvořit s žáky změnou parametrů podobná cvičení. Cvičení vytvořené k této lekci jsou doporučená podle návaznosti na předchozí výklad. Odkazy k jednotlivým cvičením k této lekci najdeme zde: <http://gawl.pencilcode.net/edit/LEKCE1/CVIKO/>

Tři čtverce

Naším prvním úkolem bude vytvořit tři čtverce. Začne se nejmenším čtvercem, který má délku strany 25 bodů, poté středním 50 bodů a největším o délce 100 bodů. Tyto čtverce budou mít stejnou tloušťku obrysu pera, ale jinou barvu.



Obrázek 25-Tři čtverce v sobě

Kříž

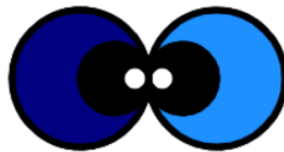
Úkolem s názvem „Vybarvený kříž“ si zopakujeme vyplnění obrazce barvou, se kterým jsme se seznámili v první lekci. Pro tvorbu kříže použijeme příkazy **fd** a **rt/lt**. Následně obrazec vybarvíme příkazem **fill**.



Obrázek 26-Kříž

Očička

Dalším úkolem je obrazec, který se skládá z šesti vybarvených kružnic. První dvě největší kružnice jsou stejně velké s modrou výplní (**fill**) a nakreslené černou barvou pera (**pen**). Uvnitř obou největších kružnic se nachází dvě střední kružnice. Tyto střední kružnice mají poloviční poloměr než kružnice největší. Obě střední kružnice jsou následně vybarvené černou barvou (**fill**). Na závěr vytvoříme dvě mini kružnice, které budou nakreslené bílou barvou pera (**pen**) a vybarvené bílou barvou (**fill**).



Obrázek 27-Očička

Domeček

Dalším úkolem bude domeček jedním tahem, bez překříženého čtverce, tudíž jde o snadnější variantu.

Nápověda: Želvička začne v pravém či levém horním rohu a vytvoří čtverec o délce strany 100 bodů. Bude potřeba spočítat odvěsny rovnoramenného trojúhelníka (zatočení bude 45 stupňů a délka jedné strany rovnoramenného trojúhelníka bude 70 bodů).



Obrázek 28-Domeček jedním tahem

4.4 Lekce 2 – Základy

Vstupní požadavky na žáky:

- Žáci se orientují v bance příkazů.
- Zvládají přetahování příkazů na dané místo ve scénáři (**fd**, **rt/lt**, zatočení o úhel **rt/lt** a poloměr otáčení **r**, **pen**, **fill**).
- Žáci vědí, co jednotlivé příkazy znamenají, jak a k čemu se používají.
- Žáci zvládají kopírovat příkazy ve scénáři a vkládat na určené místo.

Nové dovednosti:

- Žáci se seznámí s příkazy **dot** a **box** z kategorie **art**.
- Žáci se seznámí s příkazem **bk** (jít pozadu) z kategorie **move**.

Tečky různými barvami

Tečka je jeden z hlavních tvarů, který budeme v této lekci využívat. Jedná se o vyplněný kruh, jemuž měníme velikost a barvu.

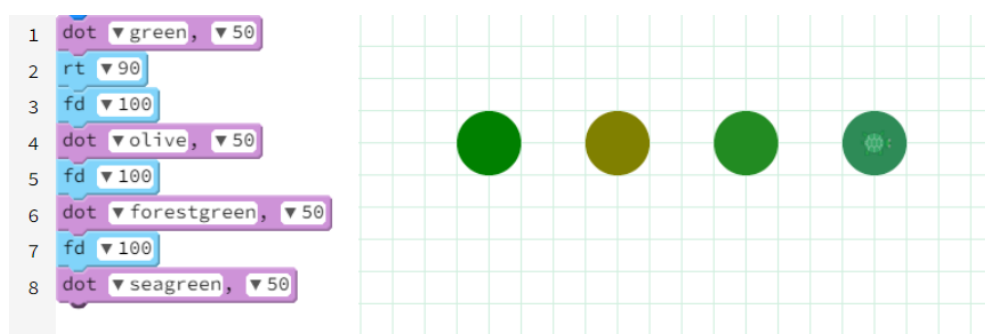


Obrázek 29-Tečky různými barvami

Příkaz na vytvoření tečky se nachází opět v kategorii **art**, jmenuje se **dot**. Přetáhneme si ho do scénáře a spustíme program, abychom viděli, jak tečka vypadá. Změníme ji barvu a dáme ji velikost 50. Budeme chtít čtyři tečky vedle sebe. Použijeme příkaz otočení doprava

(**rt**) a následně popojdeme dopředu o 100 bodů (**fd 100**). Vložíme příkaz další tečky a opět příkaz **fd**. Následně opakujeme ještě dvakrát pro nakreslení čtyř teček vedle sebe.

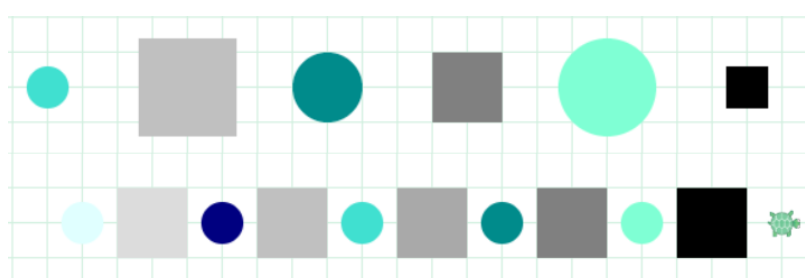
Může nastat problém, kdy žáci vloží příkaz **pen** do scénáře a při přesunu pomocí **fd** želva zanechá stopu. Lze ošetřit odstraněním příkazu **pen**.



Obrázek 30-Hotové řešení

Mini úkol: Místo příkazu pro vytvoření tečky (**dot**) si zkusíme přetáhnout příkaz **box** z kategorie **art**. Příkaz **box** nám nakreslí vyplněný čtvereček. Můžeme mu také měnit velikost a barvu. Horizontální sled tvarů změňme na vertikální (na začátku scénáře odebereme příkaz **rt**).

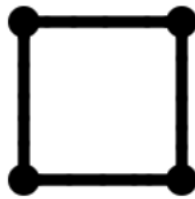
Nápověda: Vyzkoušíme si nakreslit na přeskáčku tečku (**dot**) a čtvereček (**box**). Buď necháme stejnou velikost a pouze změňme barvu obou tvarů, nebo zkusíme pozměnit velikost a barvu obou obrazců (viz Obrázek 31).



Obrázek 31-Mini úkol k tečkám

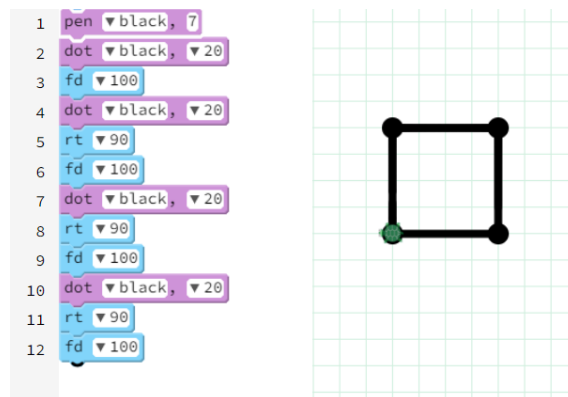
Tečko-čtverec

Zkusíme aplikovat nově naučené tečky na vrcholy čtverce, jak je patrné v Obrázek 32.



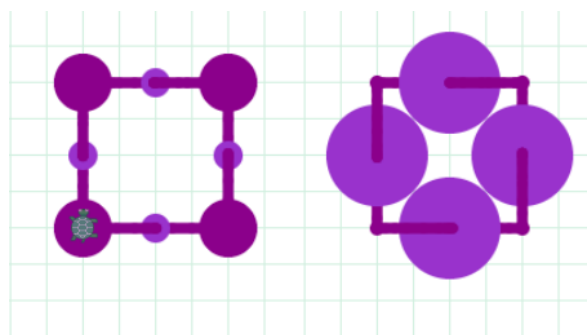
Obrázek 32- „Tečko-čtverec“

Čtverec už vytvořit umíme. Začneme tedy přetáhnutím příkazu **pen** z kategorie **art**. Dále pro vytvoření prvního vrcholu tečkou přetáhneme příkaz **dot**. Nadefinujeme mu stejnou barvu jako barvu pera. Poté použijeme příkaz **fd** s hodnotou 100 na vytvoření strany čtverce. Nyní je vhodná doba zopakovat si kopírování příkazů. Nejdříve si označíme příkaz **dot** a **fd** přejetím myši a držením levého tlačítka myši. Příkazy nám zmodrají a poté označené příkazy zkopírujeme klávesovou zkratkou **Ctrl + C**. Vložíme třikrát pod námi zkopírované příkazy (**dot** a **fd**) klávesovou zkratkou **Ctrl + V**. Pro zatočení použijeme příkaz **rt** a budeme ho vkládat mezi příkazy **dot** a **fd**. Buď pokračujeme klasicky v přetahování příkazů z banky nebo používáním kopírování příkazů.



Obrázek 33-Hotové řešení

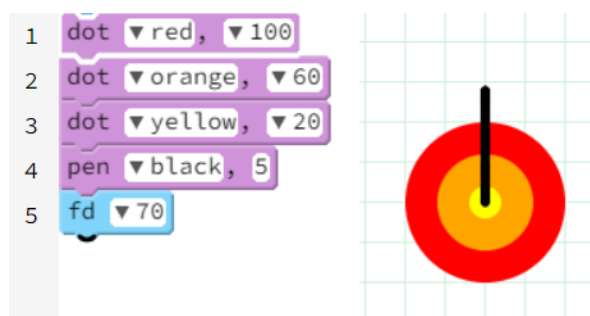
Mini úkol: Vytvořený „tečko-čtverec“ si ponecháme a vyzkoušíme si rozdělit strany čtverce na půl, abychom mohli do každé strany vložit tečku. Náповěda: Místo jednoho příkazu **fd** použijeme dva s polovičními hodnotami našeho předešlého **fd**. Mezi tyto dva příkazy vložíme příkaz **dot** pro vytvoření tečky, která bude odlišnou barvou a velikostí k tečkám na vrcholech čtverce. (viz Obrázek 34).



Obrázek 34-Mini úkol k "tečko-čtverci"

Terč

Dalším obrazcem pro procvičení příkazu **dot** bude terč. Na tomto příkladu také zjistíme, že ve výstupu dochází k překrývání jednotlivých příkazů vykonaných želvičkou. To znamená, že čím novější je příkaz ve scénáři (je níže), tím se ve výstupu námi požadovaný obrazec (tvar) dostane do popředí (je nad ostatními), jak ukazuje Obrázek 35Obrázek 35.



Obrázek 35-Překreslování příkazů

Zaměřovací kříž uděláme pomocí dvou příkazů. Příkazem **fd**, který už známe a novým příkazem **bk** (move backward) neboli jít pozadu.

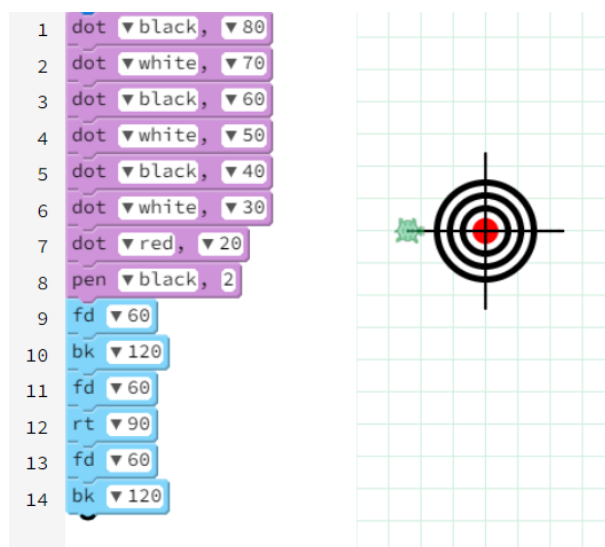


Obrázek 36-Terč

Začneme přetáhnutím příkazu **dot**, nastavíme černou barvu a velikost 80 bodů. Následně přetáhneme opět příkaz **dot**, nastavíme mu bílou barvu a zmenšíme na 70 bodů. Spustíme program, abychom viděli efekt překreslení černé tečky bílou tečkou. Opět přetáhneme černou a bílou tečku, které budeme zmenšovat o 10 bodů. Takhle budeme pokračovat do bílé tečky o velikosti 30 bodů. Střed uděláme červenou barvou a velikostí 20.

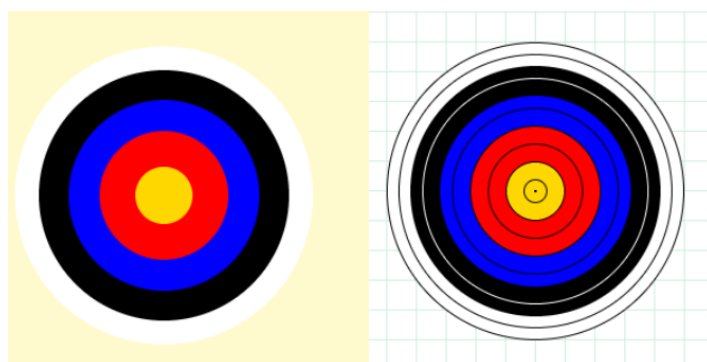
Na tvorbu zaměřovacího kříže budeme potřebovat příkaz **pen** o velikost 2. Poté přetáhneme příkaz **fd** a nakreslíme čáru o 60 bodech. Spustíme program, abychom viděli, kam se nám želvička dostala. Následně přetáhneme příkaz **bk**, jehož hodnota bude dvojnásobná hodnoty příkazu **fd**. Tím vytvoříme vertikální čáru zaměřovacího kříže. Pro přesunutí na střed musíme opět využít příkaz **fd** 60 bodů. Želvička se nachází na středu, otočíme ji o 90 stupňů a pokračujeme příkazem **fd** a **bk**.

Dalším řešením pro tvorbu zaměřovacího kříže je pohyb želvičky směrem rovně, následně nám couvne o stejný kus zpátky a natočí se směrem doprava. Celé se to zopakuje čtyřikrát.



Obrázek 37-Hotové řešení

Mini úkol: Žáci si najdou na internetu libovolný obrázek terče pro lukostřelbu a pokusí se jej nakreslit přesně podle předlohy, kterou vyhledali. (viz Obrázek 38).



Obrázek 38-Mini úkol k terči

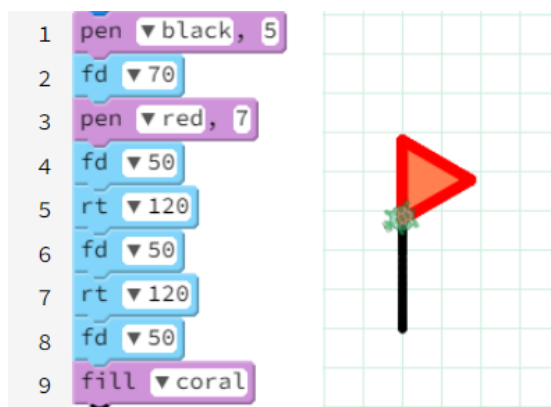
Vlajka

Dalším úkolem bude vlajka, pomocí níž se žáci naučí postupně spojovat více barevných čar a tvarů dohromady.



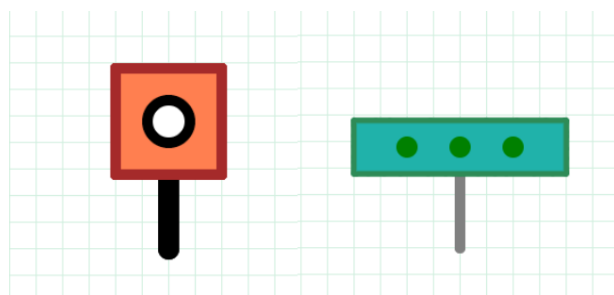
Obrázek 39-Vlajka

Začneme příkazem **pen** a pomocí **fd** vytvoříme čáru dlouhou 70 bodů. Spustíme program, abychom viděli výstup. Přetáhneme další příkaz **pen** a změním barvu pera na červenou a tloušťku obrysu pera nastavíme o něco větší. Poté začneme vytvářet vlajku, která se skládá ze tří stran. Příkazem **fd** nakreslíme jednu stranu vlajky. Zatočíme o 120 stupňů doprava (**rt**) a nakreslíme další stranu. Pro vytvoření třetí strany opět zatočíme doprava o stejný úhel a dokreslíme třetí stranu. Na následné vybarvení vlajky použijeme příkaz **fill** z kategorie **art** a vybarvíme světlejší barvou.



Obrázek 40-Hotové řešení

Mini úkol: Jako další úlohu vytvoříme, namísto červené vlaječky na tyči, čtvereček s tečkou či obdélník s puntíky.



Obrázek 41-Mini úkol ke vlaječce

Nápověda k prvnímu tvaru: Stranu čtverce, která leží na tyčce, začneme kreslit od jejího středu. Po dokončení nakreslení čtverce, čtverec vybarvíme příkazem **fill**. Pomocí příkazu **pu** odebereme pero, přesuneme se doprostřed čtverce a uděláme tečku.

Nápověda k druhému tvaru: Stranu obdélníka, který leží na tyčce, začneme kreslit od jeho středu. Po jeho nakreslení ho opět vybarvíme (**fill**). Přesuneme se doprostřed, kde uděláme puntík (**dot**). Další puntík uděláme na levou a pak na pravou stranu (viz Obrázek 41).

Trojúhelník uvnitř kruhu

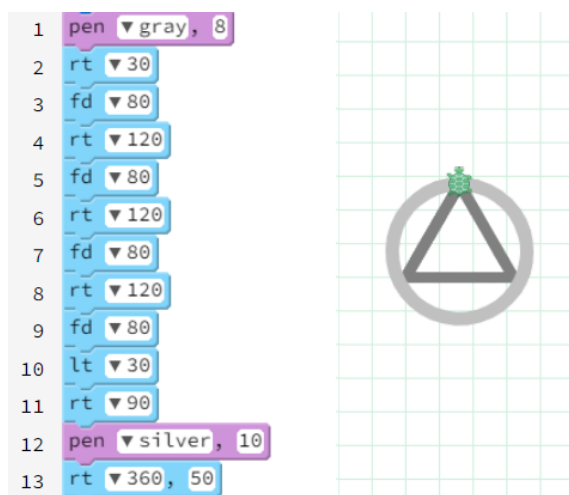
Dalším tvarem bude trojúhelník, který se nachází uvnitř kruhu, jak ilustruje Obrázek 42.



Obrázek 42-Trojúhelník uvnitř kruhu

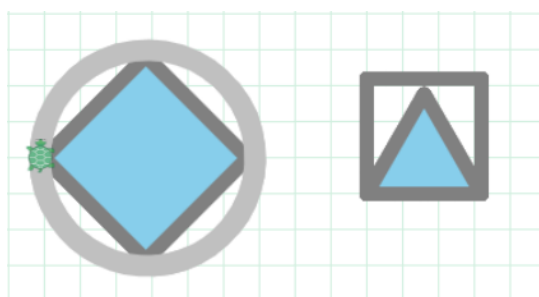
Tmavší barvou bude trojúhelník a světlejší kruh. Začneme trojúhelníkem. Natočíme želvičku o 30 stupňů doprava, aby byl výsledný trojúhelník rovně. Nakreslíme stranu dlouhou 80 bodů, vytočíme želvičku o úhel 120 stupňů a takhle zopakujeme ještě dvakrát. Zpátky želvičku natočíme o 30 stupňů doleva, aby nám koukala směrem nahoru. Spustíme program, abychom viděli náš vytvořený trojúhelník.

Na kruh si vezmeme světlejší barvu pera a vytvoříme ho příkazem `rt 360` stupňů s poloměrem 50 bodů. Před jeho vytvořením musíme želvičku vytočit doprava o 90 stupňů (`rt`), aby nám kružnice přesně sedla na trojúhelník.



Obrázek 43-Hotové řešení

Mini úkol: Žáci mohou navázat a zkusit si vytvořit kosočtverec uvnitř kruhu anebo trojúhelník ve čtverci. Obrys kosočtverce a kruhu bude jinou barvou pera a tloušťkou obrysu pera. Trojúhelník ve čtverci můžeme ponechat stejnou barvou a tloušťkou obrysu pera. (viz Obrázek 44).



Obrázek 44-Mini úkol k trojúhelníku uvnitř kruhu

4.5 Cvičení k lekcí 2

Cvičení si vyučující vybere sám podle obtížnosti a hodinové dotace. Jednotlivá cvičení prověří zvládnutí této lekce. Vyučující může vytvořit s žáky změnou parametrů podobná cvičení. Cvičení vytvořené k této lekci jsou doporučená podle návaznosti na předchozí výklad. Odkazy k jednotlivým cvičením k této lekci najdeme zde:

<http://gawl.pencilcode.net/edit/LEKCE2/CVIKO/>

Duha z teček

Prvním cvičením k této lekci bude nakreslení duhy z teček. Základní barvy duhy jsou červená, oranžová, žlutá, zelená, modrá a fialová barva. Tečky budou stejně veliké a lehce se budou překrývat.



Obrázek 45-Duha z teček

Kruh ve čtverci s tečkami

Dalším úkolem bude kruh ve čtverci. Nejdříve nakreslíme kruh a vybarvíme ho. Okolo něj dokreslíme čtverec a přemístíme se doprostřed čtverce, kde uděláme tečky, jak je znázorněno na předloze v Obrázek 46.

Nápověda: Kružnice bude olivovou barvou a vyplněná černou. Okolo bude čtverec s větší tloušťkou obrysu. Pro přemístění se využije příkaz **fd**, ale odebereme pero (**pu**). Nakonec se vytvoří tři tečky (**dot**) uprostřed kruhu.

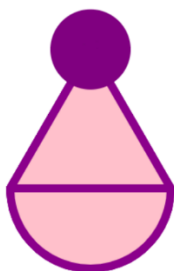


Obrázek 46-Kruh ve čtverci s tečkami

Kapka s bambulí

Další tvar, který si žáci vyzkoušejí, je možné nakreslit různými způsoby. Jedná se o půlkruh, na který navazuje trojúhelník s velkou tečkou na vrcholu.

Nápověda: Po dokončení půlkruhu a trojúhelníka se želvička bude muset vrátit nebo zopakovat trasu jedné strany trojúhelníka pro vytvoření tečky na vrcholu. Žáci ještě neznají příkazy **jump to/move to**.

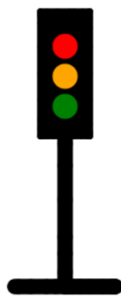


Obrázek 47-Kapka s bambulí

Semafor

Posledním cvičením k naší druhé lekci bude semafor. Jedná se o obsáhlejší cvičení, kde si zopakujeme vyplnění tvaru (**fill**), pohyb želvičky rovně (**fd**), pozpátku (**bk**) nebo také vytvoření teček (**dot**) a odebrání pera (**pu**).

Nápověda: Začneme nakreslením obdélníka a následně ho vybarvíme. Použijeme příkaz pro odebrání pera. Přesuneme se želvičkou dovnitř obdélníka a přidáme tečky podle barev semaforu. Zpátky se přesuneme příkazem **bk** na dolní stranu semaforu. Navrátíme pero příkazem **pd** a dokreslíme stojan.



Obrázek 48-Semafor

4.6 Doplnková cvičení (k lekcím 1 a 2)

Doplňková cvičení budou sloužit jako procvičení na doma nebo pro šikovnější žáky jako další cvičení v hodinách. Mají prověřit zvládnutí lekce: Základy 1 a 2. Cvičení jsou náročnější a vyžadují více času a přemýšlení o úlohách. Odkazy k doplňkovým cvičením 1: <http://gaw1.pencilcode.net/edit/DOP1/>

Smajlík

Tvar „Smajlík“ vytvoříme pomocí jedné větší tečky. Následně využijeme příkazy **fd** a **bk**, kterými se budeme přemísťovat, abychom nakreslili oči a ústa (**dot**, úhel zatočení **rt** a poloměr otáčení **r**).



Obrázek 49-Smajlík

Lod'ka

Dalším doplňkovým úkolem bude lod'ka jedním tahem, jak je vidno na Obrázek 50. Pomocí příkazu **fd** a úhlem zatočení **rt** s poloměrem otáčení **r**, vytvoříme tělo lodi, které vybarvíme modrou. Následně vytvoříme stožár a vlaječku, kterou vybarvíme červenou barvou.



Obrázek 50-Lodka jedním tahem

Opičák

Další úkol bude vytvořit opičku. Začneme příkazem **dot** pro vytvoření ouška. Přidáme mezeru pomocí **fd** a následně vytvoříme další ouško. Přesuneme se doprostřed mezi ouška a pomocí tečky nakreslíme hlavu. Příkazy **fd** a **bk** (pro přesouvání), **dot** a úhel zatočení **rt** s poloměrem otáčení **r** dokreslíme opičákovi oči a pusy.



Obrázek 51-Opičák

YinYang

YinYang vytvoříme pomocí dvou půlkružnic (příkazy **rt** o poloměr otáčení) a následně pro ohraničení velkou půlkružnicí (opět příkaz zatočení **rt** s poloměrem otáčení **r**) černou barvou. Následně vyplníme bílou barvou pomocí příkazu **fill**. To samé zopakujeme u spodní černé části. Poté se přemístíme příkazy **fd** a uděláme tečky (**dot**).



Obrázek 52-YinYang

BMW

Další úkol je znak BMW, který nakreslíme pomocí černé a bílé tečky (**dot**). Následně vytvoříme dvě čtvrtky kruhu šedou barvou příkazem **fd** a příkazem zatočení o úhel **rt** s poloměrem **r**. Poté je vybarvíme modrou barvou.



Obrázek 53-BMW

4.7 Lekce 3

Vstupní požadavky na žáky:

- Zvládají přetahování příkazů na dané místo ve scénáři a tvoření složitějších obrazců.
- Orientují se ve výstupu programu a zvládají pozměnit program dle dalších instrukcí a různých předloh jiných tvarů a obrazců.
- Pracují se základními příkazy z kategorie **move** (**fd**, **bk**, **rt/lt**, **rt/lt** s poloměrem otáčení **r**)
- Pracují se základními příkazy z kategorie **art** (**dot**, **box**, **pen**, **fill**, **pu** a **pd**).
- Žáci zvládají kopírovat příkazy ve scénáři a vkládat na určené místo.

Nové dovednosti:

- Osvojení si příkazů **jumpto**, **moveto** a **home** z kategorie **move** pro přemístění po souřadnicích.
- Žáci se naučí nový příkaz **speed** z kategorie **move** pro zrychlení pohybu želvičky ve výstupu.

Výklad k souřadnicím

Želvička se umí přemísťovat po souřadnicích pomocí kurzoru myši a příkazům k tomu určeným. Na začátku je nastavená na souřadnicích [0:0]. Příkazem **jumpto** želvička skočí

na určitou souřadnici bez toho, aby zanechala stopu pera. Příkazem **moveto** se želvička posune přímým směrem na určitou souřadnici s tím, že zanechá stopu pera. Příkazem **home** se želvička navrátí na výchozí pozici a to souřadnice [0:0]. Příkaz **home** také navrací želvičku při otáčení do výchozí pozice 0° (tedy bude vždy opět koukat směrem nahoru).

Jako příklad pro vysvětlení jsem zvolila Berušku. Nakreslíme si s dětmi černou tečku (**dot**), která bude značit hlavu a poté větší červenou tečku (**dot**) pro tělo berušky. Poté dětem ukáží příkaz **jump to**, díky kterému dokreslíme tečky různé velikosti a na různé místa v těle berušky (viz Obrázek 54).



Obrázek 54-Beruška

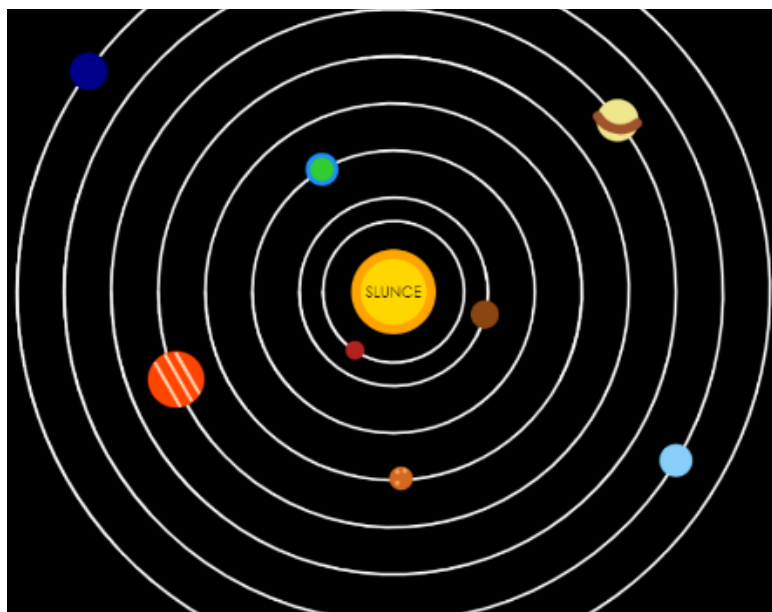
Cvičení k souřadnicím

Pro procvičení práce se souřadnicemi jsem vybrala sluneční soustavu. Žáci dostanou podklad, a to nakreslené orbity, na které pomocí příkazů **jump to** budou dokreslovat planety. Žáci si najdou na internetu obrázek, jak jsou planety seřazené, jakou mají velikost a jak vypadají, podle toho se je budou snažit dokreslit na podklad.

Žáci si vyberou oběžnou dráhu, poté přejedou myší a vpravo dole se jim ukáží souřadnice kurzoru. Souřadnice si opíšou do příkazu **jump to** a mohou pokračovat v kreslení planety. Takto postupují dále. Pro zrychlení vykreslování planet mohou žáci použít nový příkaz **speed**, který se nachází v kategorii **move**. Tento příkaz přetáhnu kamkoliv do scénáře. Díky tomu zjistí, že všechny ostatní příkazy za příkazem **speed** se začnou vykonávat rychleji. Příkazu mohou nastavit jakoukoliv rychlost, doporučuje se ponechávat rychlost 10. Příkaz se většinou přetahuje pouze jednou, a to na úplný začátek programu. Existuje také

příkaz **speed infinity**. Tento příkaz nakreslí ihned náš program, a to bez viditelného pohybu želvičky.

Pro přehlednost si žáci každou planetu uloží do příkazu **button**, který se nachází v kategorii **control**. Jedná se o tlačítko, díky kterému dané planety vykreslí postupně po jeho stisknutí.



Obrázek 55-Příklad hotového řešení

Hotové řešení žáků může být zpracováno různými způsoby, a to jak barevným, tak umístěním planet na oběžné dráhy.

4.8 Doplnková cvičení (k lekcím 1, 2 a 3)

Doplnková cvičení budou sloužit jako procvičení na doma nebo pro šikovnější žáky jako další cvičení v hodinách. Mají prověřit zvládnutí další lekce ke stávajícím základům. Cvičení jsou náročnější a vyžadují více času a přemýšlení o úlohách. Odkazy k doplnkovým cvičením 2: <http://gawl.pencilcode.net/edit/DOP2/>

Mimoň

Mimoň je nakreslen pomocí všech dosud naučených a probraných příkazů. Jedná se o komplexní cvičení, které prověří žákovy schopnosti a porozumění práci v prostředí. Jde o složitější práci se souřadnicemi spojenou s přemísťováním a natáčením želvičky.



Obrázek 56-Mimoň

Sněhulák

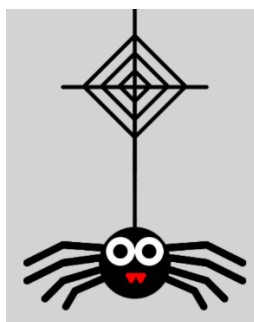
Tělo sněhuláka je tvořeno třemi příkazy pro tvorbu tečky (**dot**), poté se želvička přemísťuje po souřadnicích příkazy **jump to**. Dále nakreslíme příkazy **dot** ruce, knoflíky, oči a nos. Následně se želvička přemísťí vedle těla sněhuláka dolů a pomocí příkazů (**pen** a **moveto**) nakreslí hůl. Pomocí příkazu **jump to** skočí želvička nad hlavu sněhuláka a příkazy (**fd** a zatočení **rt** s poloměrem **r**) vytvoří pokrývku hlavy.



Obrázek 57-Sněhulák

Pavouk

Pavouk se skládá ze dvou částí. První částí je nakreslení pavučiny a druhou nakreslení pavouka. Pavučinu začneme křížem, poté pomocí **moveto** spolu s **jump to** nebo **jump to** spolu s **fd** ji dokreslíme. Tělo pavouka následně vytvoříme z tečky (**dot**). Dokreslíme oči a kusadla. Nožky pavouka kreslíme pomocí **moveto**, je to nejjednodušší způsob. Pomocí **fd** a **rt/lt** by to bylo o dost složitější, protože bychom museli počítat se složitějšími úhly a souřadnicemi.



Obrázek 58-Pavouk

4.9 Lekce 4 - Cykly

Vstupní požadavky na žáky:

- Orientují se s přehledem v prostředí Pencil Code i v kategoriích příkazů.
- Umí používat příkazy k přemístění želvičky (prostředníka pro vykonávání příkazů).

Nové dovednosti:

- Seznamují se s cykly z kategorie **control** a jejich následným využití v programu.
- Seznamují se s proměnnými.
- Žáci efektivněji kreslí základní geometrické tvary pomocí cyklů (čtverec, trojúhelník).

Čtverec pomocí cyklu

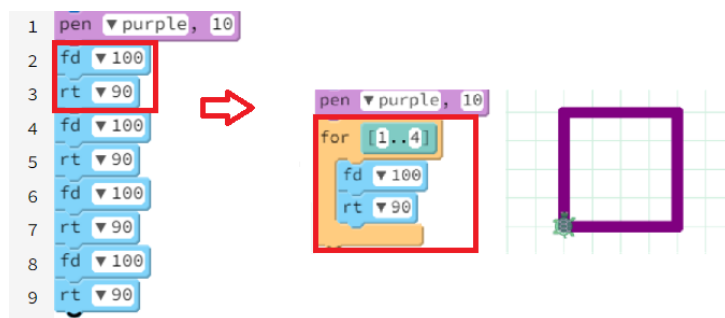
Čtverec nám je už znám v mnoha podobách. Pokusíme se ho proto nakreslit efektivně, abychom neměli velké množství shodných příkazů za sebou. K tomu využijeme nové příkazy pro cykly. Cyklus v programování slouží k opakování skupiny příkazů.



Obrázek 59-Čtverec pomocí cyklu

Nejprve si žáci otevrou první program z první lekce s názvem čtverec. Následně dojdou ke zjištění, že se příkazy **fd 100** a **rt 90** opakují. Dvojice **fd** a **rt** značí jednu stranu čtverce.

Tudíž použijeme cyklus, který se nachází v kategorii **control**. Než začneme s cyklem, přetáhneme si příkaz **pen**. Pro začátek si vyzkoušíme cyklus **for**, do kterého vložíme příkazy pro nakreslení jedné strany čtverce (**fd** 100 a **rt** 90). Jelikož chceme 4 strany, tak nastavíme opakování 1 až 4. Spustíme program a následně uvidíme vykreslení čtverce pomocí cyklu **for** (viz Obrázek 60).



Obrázek 60-Hotové řešení

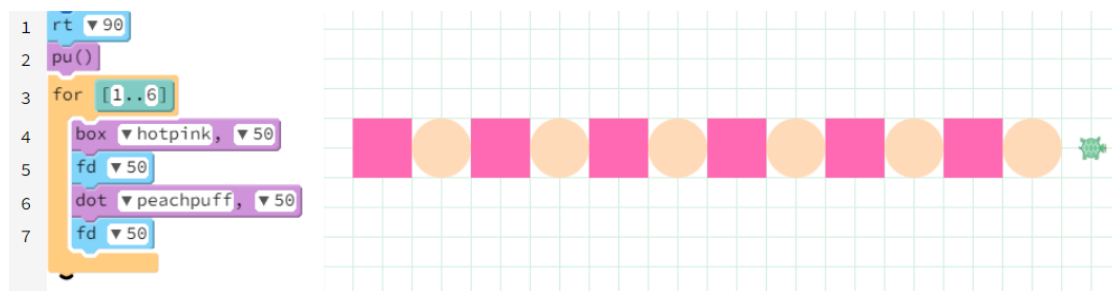
Tvary

Dalším tvarem bude had tvořený tečkami a kostičkami, které jsou vykreslené bezprostředně za sebou.



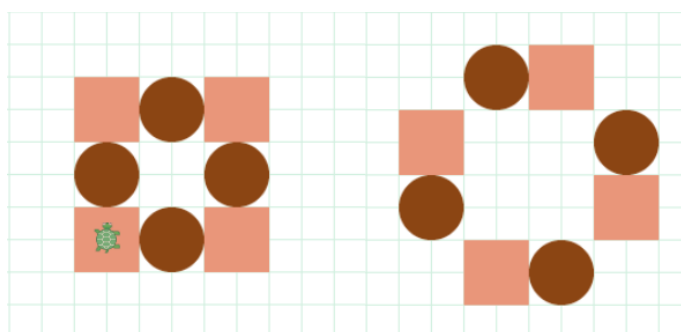
Obrázek 61-Tvary v cyklu

Nejdříve si želvičku natočíme směrem doprava příkazem **rt**. Následně vložíme opět cyklus **for** pro opakování příkazu. Budeme jej chtít opakovat šestkrát. Do cyklu poté přetáhneme **box** s růžovou barvou a velikostí 50. Příkazem **fd** popojdeme taktéž o 50 bodů, aby se nám další tvar nepřekrýval anebo nevznikla mezera mezi tvary. Vložíme příkaz pro nakreslení tečky (**dot**) s jinou barvou a velikostí 50 a opět popojdeme. Žáci si mohou zkusit nejdříve vložit jen kostičku (**box**) a popojít dopředu a poté navázat s tečkou (**dot**). Takto žáci vše vidí, a uvědomí si, jak fungují příkazy v cyklu



Obrázek 62-Hotové řešení

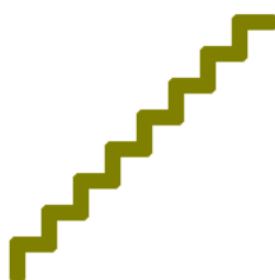
Mini úkol: Hada zkusíme zatočit a vytvořit z něj čtverec nebo kruh (viz Obrázek 63). Nápověda: cyklus tedy opakujeme čtyřikrát, a nakonec těla cyklu vložíme příkaz zatočení doprava (**rt**) o úhel 90 stupňů.



Obrázek 63-Mini úkol k hadovi

Schody

Další úkol bude nakreslit schodiště ze stejných osmi schodů.

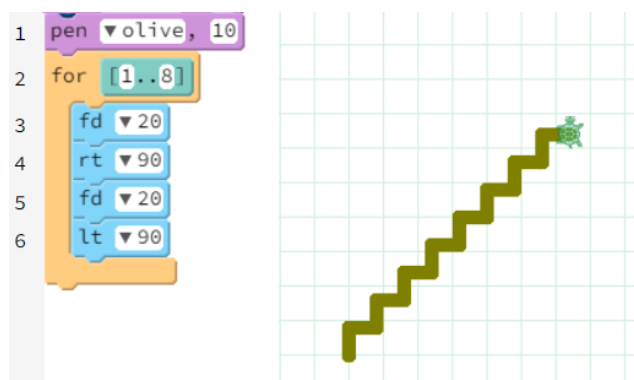


Obrázek 64-Schody

Opět se jedná o obrazec, ve kterém se něco stejného opakuje, proto je vhodné použít cyklus. Nejdříve vložíme příkaz pro kreslení, ten můžeme vložit buď před cyklus, nebo dovnitř cyklu, je to na osobní volbě jednotlivce. Následně vložíme cyklus **for**, který se nachází v

kategorii **control**. Chceme nakreslit osm schodů, tudíž se nám cyklus bude opakovat osmkrát.

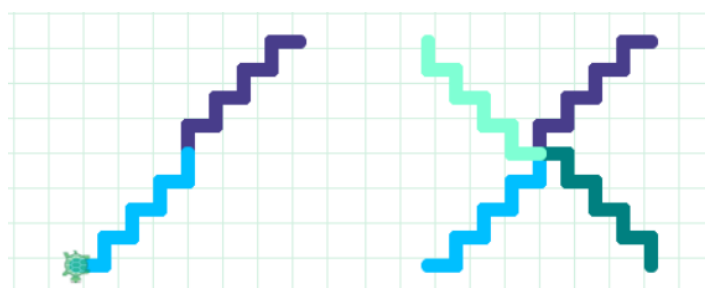
Začneme příkazem pro pohyb želvičky (**fd**) třeba o 20 bodů. Dále vložíme příkaz pro zatočení doprava (**rt**). Spustíme program, abychom viděli předběžně, co nám želvička vykreslí. Vidíme, že potřebujeme ještě jednu příkaz pro pohyb želvičky (**fd**) a poté příkaz pro zatočení vlevo (**lt**), abychom měli kompletní schod. Spustíme program a máme schodiště o osmi schodech.



Obrázek 65-Hotové řešení

Mini úkol: K našim schodům si vyzkoušíme vytvořit schody dolů a jinou barvou. Přemístíme se a zkusíme vytvořit schody, které se budou křížit (viz Obrázek 66).

Nápověda: Pro přemístění je vhodné využít **jump to** či **home** příkazy.



Obrázek 66-Mini úkol ke schodům

Desetiúhelník

Posledním úkolem bude vytvoření desetiúhelníku. V našem případě má desetiúhelník deset stejně dlouhých stran a deset stejně velkých vnitřních úhlů.

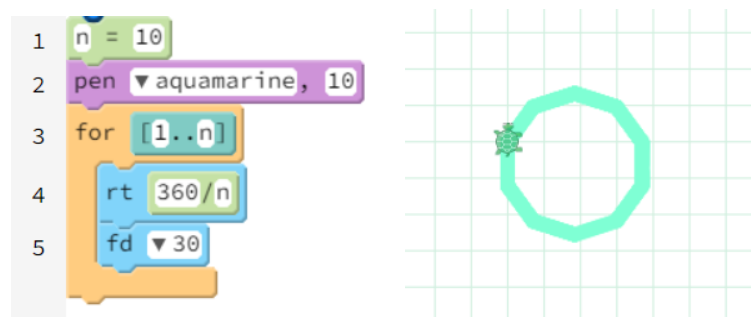


Obrázek 67-Desetiúhelník

Jedná se o pravidelný tvar s opakující se stejnou délkou strany a stejným úhlem, proto opět použijeme cyklus **for**. V předešlých úlohách jsme do příkazů, psali číselné výrazy napevno. V této úloze si zkusíme ukázat použití proměnných, které využijeme při opakování výrazů v programu.

Proměnné najdeme v kategorii **operators**. Jedná se hned o první příkaz se vzorem **x=10**. Přetáhneme ho do scénáře a pojmenujeme **n**. Do proměnné **n** uložíme číslo 10, nutné pro desetiúhelník. Dále přejdeme do kategorie **control** a vložíme cyklus **for**. Hlavičku cyklu **for** přepíšeme na 1 až **n** opakování. Uvnitř cyklu budeme potřebovat příkaz pro kreslení perem (**pen**). Potom vložíme příkaz pro zatočení doprava (**rt**) a hodnotu přepíšeme na $360/n$. Za tento příkaz vložíme příkaz pro pohyb želvičky (**fd**), tím nakreslíme stranu desetiúhelníku a spustíme program. Ve výstupu uvidíme náš výsledný desetiúhelník.

Poznámka: Hodnotu pro otáčení doprava (**rt**) určíme díky tomu, že se želvička otočí právě jednou dokola (o 360°). Jelikož chceme nakreslit pravidelný desetiúhelník (v každém vrcholu se želvička otočí o stejný úhel), proto je výsledná hodnota $360/10$.

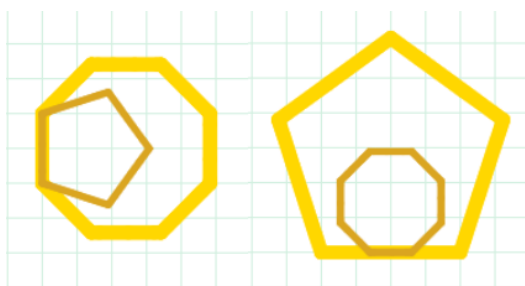


Obrázek 68-Hotové řešení

Pro procvičení a pochopení proměnných a cyklů, si žáci vyzkouší měnit hodnoty proměnné svého programu. Díky proměnným nám stačí pouze jedna editace hodnoty proměnné (na

začátku programu). Ušetří nám to ruční úpravy v rozsáhlejších programech, kde je vyšší pravděpodobnost, že na nějakou změnu zapomeneme.

Mini úkol: Žáci dostanou za úkol vytvořit dva obrazce. První bude osmiúhelník a uvnitř něj bude pětiúhelník a druhý obrazec bude právě opačný (viz Obrázek 69). Na to budou potřebovat vytvořit další cyklus a vložit jej pod první cyklus, dále změnit název proměnné, barvu obrazce a počet stran.



Obrázek 69-Mini úkol k desetiúhelníku

4.10 Cvičení k lekci 4

Cvičení si vyučující vybere sám podle obtížnosti a hodinové dotace. Jednotlivá cvičení prověří zvládnutí této lekce (především pochopení cyklu **for**). Vyučující může vytvořit s žáky, pomocí změn parametrů, podobná cvičení. Cvičení vytvořená k této lekci jsou doporučena podle návaznosti na předchozí výklad. Odkazy k jednotlivým cvičením k této lekci najdeme zde: <http://gawl.pencilcode.net/edit/LEKCE4/CVIKO/>

Trojúhelníky

Dva trojúhelníky žáci nakreslí do jednoho programu. Pomocí dvou cyklů **for**, které budou za sebou. Žáci začnou příkazem pro kreslení perem (**pen**). Poté vloží příkaz pro tvorbu cyklu **for** z kategorie **control**. Následně přidají příkaz pro pohyby želvičky (**fd**) a příkaz pro zatočení doprava (**rt**). Cyklus bude mít tři opakování, tímto se nakreslí první trojúhelník. Následně žáci vloží za cyklus příkaz zatočení doleva (**lt**), který je nutný pro nakreslení druhého trojúhelníka. Dále zkopírují svůj cyklus **for**, změni barvu na **pink** a takto mají vytvořeny dva trojúhelníky z jednoho vrcholu.



Obrázek 70-Trojúhelníky

Pila

Pilu vytvoříme opět pomocí cyklu **for**. Pila má šest zubů, což znamená, že se cyklus bude opakovat šestkrát. Před cyklus zařadíme ještě příkaz pro kreslení (**pen**) a příkaz zatočení doprava (**rt**), protože nám pila směřuje doprava.

Nápověda: Uvnitř cyklu budou poté příkazy zatočení vlevo (**lt**) o 45 stupňů, pohyb vpřed (**fd**), zatočení doprava (**rt**) o 90 stupňů, pohyb vpřed (**fd**) a zatočení doleva (**lt**) o 45 stupňů



Obrázek 71-Pila

Čtverec s tečkami

Dalším úkolem bude vytvořit nám už známý „tečko-čtverec“ pomocí cyklu **for**. Začneme přetažením příkazu pro kreslení (**pen**), poté z kategorie **control** cyklus **for**. Cyklus se bude opakovat čtyřikrát a uvnitř cyklu budou příkazy pro pohyb vpřed (**fd**), tečku (**dot**) a zatočení doprava (**rt**).



Obrázek 72-Čtverec s tečkami

Audi

Znak Audi je tvořený čtyřmi stejnými kružnicemi vzdálenými o stejnou délku. Žáci začnou přetažením příkazu pro kreslení (**pen**) a cyklu **for**. Uvnitř cyklu se bude nacházet příkaz pro vytvoření kružnice (**rt** 360, s poloměrem otáčením **r**), následně zatočí doprava (**rt**). Odeberou želvičce pero příkazem (**pu**) a přesunou se vpřed příkazem (**fd**). Poté musí želvičce navrátit zpátky pero příkazem (**pd**). Zatočit doleva příkazem (**lt**), tak aby želvička koukala opět směrem nahoru. Cyklus bude mít čtyři opakování.



Obrázek 73-Audi

4.11 Doplnková cvičení (k lekci 4)

Doplňková cvičení budou sloužit jako procvičení na doma nebo pro šikovnější žáky jako další cvičení v hodinách. Mají prověřit zvládnutí další lekce ke stávajícím základům. Cvičení jsou náročnější a vyžadují více času a přemýšlení o úlohách. Odkazy k doplňkovým cvičením 3: <http://gaw1.pencilcode.net/edit/DOP3/>

Vločka

Vločku nakreslíme pomocí cyklu. Cyklus se bude opakovat šestkrát. Nejdříve začneme příkazem pro kreslení perem (**pen**). Uvnitř cyklu budou poté příkazy pro pohyb želvičky vpřed (**fd**), vytvoření tečky (**dot**), pohyb vzad (**bk**) a zatočení želvičky o 360/6 stupňů doprava (**rt**), protože má vločka šest výběžků.



Obrázek 74-Vločka

Hvězda

Hvězdu vytvoříme pomocí cyklu. Cyklus se bude opakovat pětkrát. Nejdříve začneme příkazem pro kreslení perem (**pen**). Uvnitř cyklu budou příkazy pro pohyb želvičky vpřed (**fd**) a zatočení doprava (**rt**) o úhel 144 stupňů.



Obrázek 75-Hvězda

Hvězda2

Hvězdu2 vytvoříme opět pomocí cyklu. Cyklus se bude opakovat šestkrát, protože kytka je tvořena ze šesti stejných kosočtverců. Uvnitř cyklu se budou nacházet příkazy pro vytvoření jednotlivých kosočtverců. Jedná se o příkazy **fd**, **rt**, kterými dosáhneme nakreslení jednoho kosočtverce. Na konec našeho cyklu následuje příkaz pro vybarvení jednotlivých kosočtverců (**fill**).

Žáci si mohou vyzkoušet příkaz **random** z kategorie **operators**, který složí pro generování náhodných proměnných. Nadefinují mu proměnnou **color** a uvidí, že každý nakreslený kosočtverec bude vybarven náhodnou barvou.



Obrázek 76-Kytka

4.12 Lekce 5 – Funkce

Vstupní požadavky na žáky:

- Orientují se s přehledem v prostředí Pencil Code i v jeho kategoriích.
- Přemísťují želvičku, natáčejí želvičku a vytvářejí složitější útvary a obrazce.
- Užívají proměnné v programu z kategorie **operators**.
- Zvládají pracovat s cyklem **for**.

Nové dovednosti:

- Žáci se seznámí s funkcemi.
- Vytvářejí funkce a volají je v programu.
- Žáci se seznámí s příkazem **label** z kategorie **text**.
- Žáci se naučí nový příkaz **button** z kategorie **control**.

Vlajky

Funkce je logický blok kódu, který jednou napíšeme a poté ho můžeme libovolně volat, bez ohledu na to, abychom jej psali znovu a opakovali se. Příkaz pro vytvoření funkce nalezneme v kategorii **operators**. Jelikož chceme tvořit vlajky, tak si vyzkoušíme vytvořit funkci pro obdélník, kterou si nazveme **Obdelnicek**. Funkce bude obsahovat vstupní parametry (argumenty) **x**, **y**, **c**. Parametr **x** bude pro délku, **y** bude pro šířku a **c** bude pro barvu, kterou bude daný obdélník vyplněný. Uvnitř funkce poté zadáme příkaz pro pero (**pen**) s černou barvou. Následně vytvoříme **for** cyklus se dvěma opakováním pro tvorbu obdélníku. Příkaz **fd** s parametrem **x**, poté **rt** 90, opět příkaz **fd** s parametrem **y** a nakonec **rt** 90. Za cyklus zařadíme příkaz **fill** s parametrem **c** (viz Obrázek 77).

```

1 Obdelnicek = (x,y,c) ->
2   pen ▾ black
3   for [1..2]
4     fd ▾ x
5     rt ▾ 90
6     fd ▾ y
7     rt ▾ 90
8   fill ▾ c

```

Obrázek 77-Tvorba funkce *Obdelnicek*

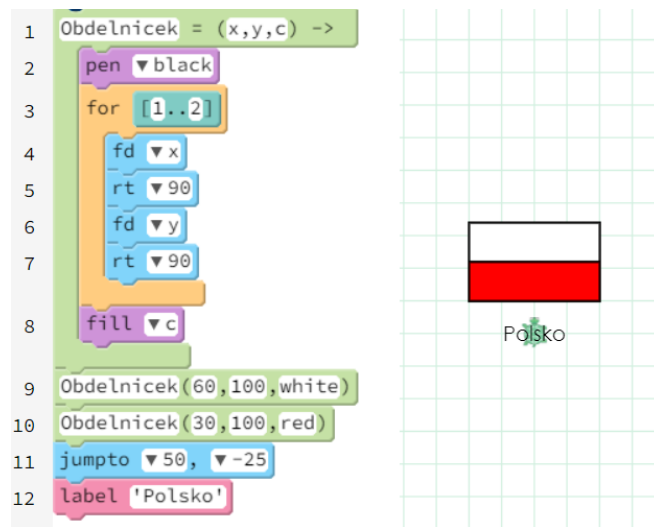
Pro odzkoušení vytvoříme vlajku Polska, která je tvořena bílým a červeným obdélníkem (viz Obrázek 78).



Polsko

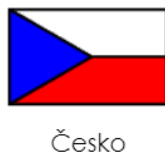
Obrázek 78-Vlajka Polska v prostředí *Pencil Code*

Funkci máme vytvořenou, nyní ji budeme volat. Příkaz pro volání funkce najdeme opět v kategorii **operators**. Vložíme ho pod naši vytvořenou funkci, opíšeme její název a do parametrů vkládáme délky stran a barvu obdélníku. Nakonec vložíme příkaz pro pojmenování výsledné vlajky, který najdeme v kategorii **text**, a nazývá se **label**. Do apostrofů napíšeme název a pomocí příkazu **jump to** název umístíme pod vlajku.



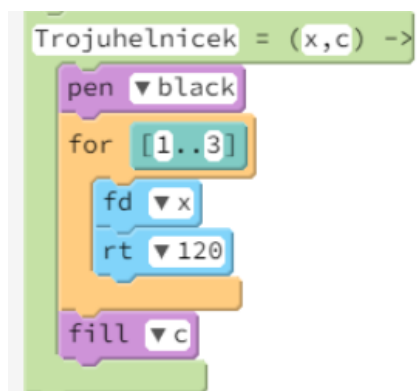
Obrázek 79-Hotové řešení

Další vlajčku, kterou si vytvoříme, bude vlajka Česka, která je tvořena ze dvou obdélníků a trojúhelníka.



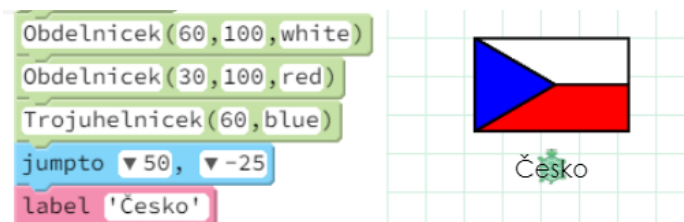
Obrázek 80-Vlajka Česka v prostředí Pencil Code

Funkci **Obdelnicek** pro tvorbu obdélníka již máme vytvořenou, takže nám zbývá vytvořit funkci pro trojúhelník. Přetáhneme si do scénáře příkaz pro funkci, kterou si nazveme **Trojuhelnicek**. Funkce bude obsahovat vstupní parametry (argumenty) **x** a **c**. Parametr **x** bude pro délku a **c** bude pro barvu, kterou bude daný obdélník vyplněný. Následně vytvoříme **for** cyklus se třemi opakováními pro tvorbu trojúhelníka. Příkaz **fd** s parametrem **x**, poté **rt** rovno 120. Za cyklus zařadíme příkaz **fill** s parametrem **c**, pro vybarvení trojúhelníka (viz Obrázek 81).



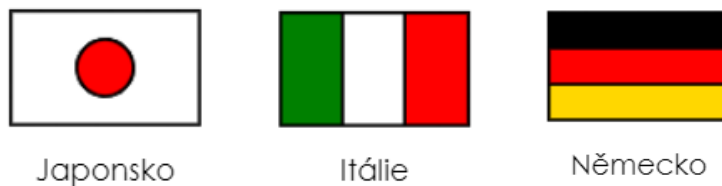
Obrázek 81-Tvorba funkce Trojuhelnicek

Funkce je vytvořena, nyní budeme volat jak funkci **Obdelnicek**, tak **Trojuhelnicek**. Příkaz pro volání funkce najdeme opět v kategorii **operators**. Vložíme ho pod naši vytvořenou funkci, opíšeme její název a do parametrů vkládáme délku strany a barvu trojúhelníka. Nakonec vložíme příkaz pro pojmenování výsledné vlajky, který najdeme v kategorii **text**, a nazývá se **label**. Do apostrofů napíšeme název a pomocí příkazu **jump to** jej umístíme pod vlajku.



Obrázek 82-Hotové řešení volání funkcí pro tvorbu vlajky Česka

Pro procvičení funkcí si žáci vytvoří funkci pro kruh a volají ho spolu s dalšími funkcemi, aby nakreslili vlajku Japonska. Také si vyzkoušejí nakreslit více pruhů vedle sebe (vlajka Itálie) nebo pod sebou (vlajka Německa).



Obrázek 83-Mini úkol k funkcím

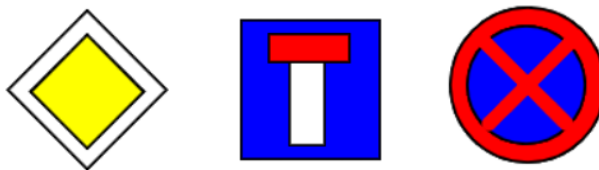
Pro přehlednost si žáci každou vlajku uloží do příkazu **button**, který se nachází v kategorii **control**. Jedná se o tlačítko, díky kterému dané vlajky vykreslí postupně po jeho stisknutí.

4.13 Cvičení k lekcí 5

Cvičení k páté lekci mají prověřit schopnosti žáků v oblasti tvorby a volání funkcí. Žáci si procvičí aplikování příkazu v různých podobách. Odkazy ke cvičením k páté lekci: <http://gawl.pencilcode.net/edit/LEKCE5/CVIKO/>

Dopravní značky

Žáci si vytvoří funkce na tvorbu dopravních značek. Učitel s žáky prodiskutuje, jaké základní tvary budou potřebovat. Bude se jednat o funkci **Kruh**, **Trojuhelník**, **Ctverec**, **Obdelník** a **Cara**. Funkce budou mít parametry (x, y a color viz vlajky). Do těchto parametrů se budou zadávat hodnoty (délky a šířky) a barva, ale až při volání jednotlivých funkcí. Žáci nakreslí několik dopravních značek (viz Obrázek 84).



Obrázek 84-Příklady dopravních značek

4.14 Lekce 6 – Klávesy

Vstupní požadavky na žáky:

- Orientují se s přehledem v prostředí Pencil Code i v jeho kategoriích.
- Užívají proměnné.
- Zvládají pracovat s cykly.
- Vytváří a volají funkce.

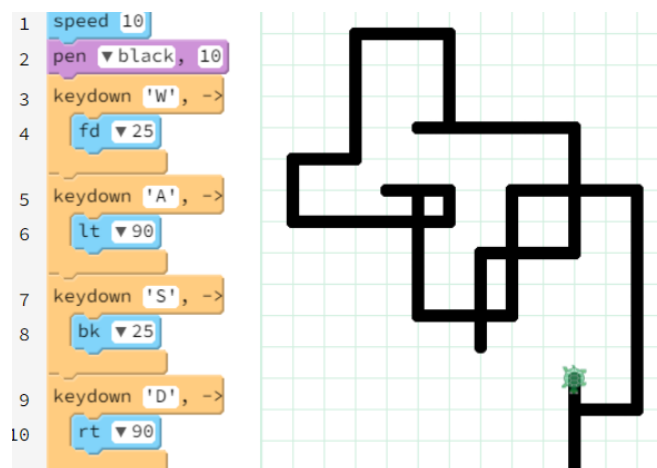
Nové dovednosti:

- Seznamují se s událostí klávesy.
- Žáci se naučí vytvářet události ke klávesám.
- Žáci se seznámí s příkazem **keydown** z kategorie **control**.

Výklad ke klávesám

V této lekci se žáci seznámí s událostí **keydown** z kategorie **control**. Do události **keydown** se přiřazují příkazy, které se vykonají po stisknutí jednotlivé klávesy. Na vyzkoušení zkusíme želvičce přiřadit klávesy pro pohyb vpřed W (**fd**), zatočení doleva A (**lt**), pohyb vzad S (**bk**) a D (**rt**) zatočení doprava.

Začneme příkazy pro zrychlení pohybu želvičky (**speed**) a kreslení (**pen**). Přejdeme do kategorie **control** a vložíme událost **keydown**. Do apostrofů vložíme název klávesy a dovnitř události příkaz pohybu (**fd**, **bk**) či zatočení (**rt**, **lt**). Takto to uděláme pro ostatní klávesy a odzkoušíme program.



Obrázek 85-Hotové řešení

Mini úkol: Žáci si zkusí k našemu programu, který už obsahuje události (pro klávesy WASD), přidat další nové události. Vytvoří událost **keydown** pro klávesy M a N. Po stisknutí klávesy M, želvička udělá tečku (**dot**) o velikosti 30 a barvě **maroon**. Po stisknutí klávesy N, udělá želvička čtvereček (**box**) o velikosti 50 a barvě **navy** (viz Obrázek 86).

4.16 Lekce 7 – Složitější konstrukce Cykly 2

Následující úlohy jsou složitějšího charakteru, proto bych je doporučila zařadit na druhý stupeň ZŠ nebo do kroužku programování.

Vstupní požadavky na žáky:

- Žáci zvládli a pochopili cyklus **for**.
- Žáci zvládají používat proměnné.

Nové dovednosti:

- Seznamují se s novým typem cyklem (**while**).
- Žáci více užívají proměnné.
- Naučí se složitější konstrukce cyklů (vnořování cyklů).

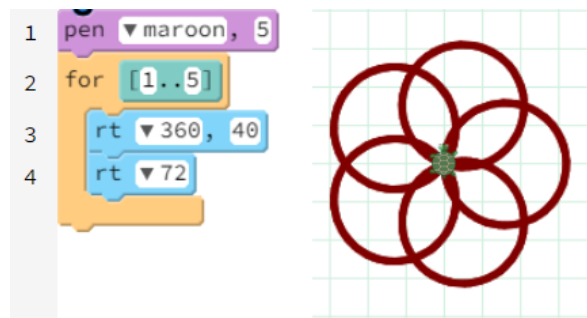
Kytička z kružnic

Na úkolu kytičky z kružnic si ukážeme její tvorbu pomocí cyklu **for**, a také pomocí cyklu **while**. Jedná se o kytičku tvořenou z pěti stejných kružnic.



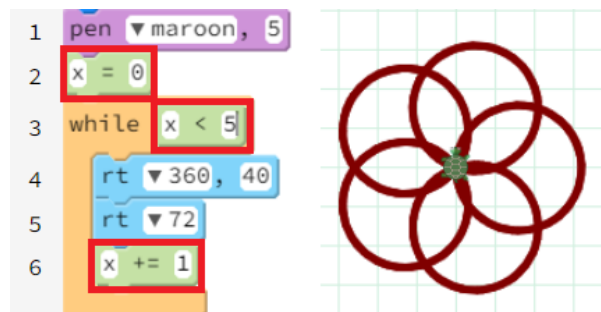
Obrázek 88-Kytička z kružnic

Nejprve si zkusíme vytvořit náš obrazec pomocí nám známého cyklu **for**. Přetáhneme si příkaz pro kreslení (**pen**) a cyklus do našeho scénáře, který se bude opakovat pětkrát. Do cyklu vložíme příkaz pro vytvoření kružnice o poloměru 40 bodů (**rt 360, 40**). Poté vložíme příkaz pro zatočení doprava (**rt**), aby nám želvička nekreslila pouze kružnice, ale tak, aby se vždy po jejich nakreslení se otočila. Aby se otočila vždy o stejný úhel a skončila opět v počáteční pozici (hledí směrem nahoru), tak opíše 360 stupňů, ale provedeme dělení pěti (máme pět kružnic). Tudíž náš výsledný příkaz bude vykonávat zatočení doprava (**rt**) o 72 stupňů.



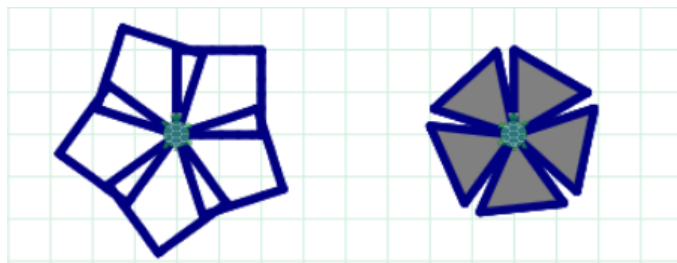
Obrázek 89-Hotové řešení cyklus for

Ukážeme si další typ cyklu, a to cyklus **while**. Nachází se opět v kategorii **control**, odkud si jej vložíme do našeho scénáře. Cyklus **while** obsahuje podmínku, která se nachází v jeho hlavičce. Podmínkou bude, že se cyklus musí opakovat, dokud náš počet kružnic nebude větší než pět. Proto si nadefinujeme proměnnou **x**, která bude mít hodnotu 0, protože začínáme počítat cykly od nuly. Hlavičku cyklu upravíme tak, aby naše požadovaná kružnice byla menší než pět vznikajících finálních kružnic tvořících kytičku. Než začneme vkládat do cyklu příkazy pro vytvoření kružnic a zatačení, vložíme dovnitř cyklu proměnnou pro zvyšování její hodnoty vždy o hodnotu jedna. Takovouto proměnnou najdeme taktéž v kategorii **operators**. Proměnná má předdefinovaný tvar **x+=1**, což znamená, že se nám naše proměnná **x**, po každém vykonání cyklu, zvětší o hodnotu jedna (například přičte k našemu číslu 0, číslo 1). Tím jsme vytvořili základ pro cyklus **while**, do kterého už můžeme vkládat příkazy z kategorie **move**. Začneme příkazem pro tvorbu kružnice s poloměrem 40 bodů (rt 360, 40). Dále vložíme příkaz pro zatačení se doprava o 72 stupňů (příkaz **rt 72** můžeme zapsat také jako 360/5). Spustíme program a máme kytičku z pěti kružnic.



Obrázek 90-Hotové řešení cyklus while

Mini úkol: Pro vyzkoušení cyklu **while** si žáci místo kružnic vytvoří pěticípou kytičku z čtverců a kytičku z trojúhelníků (viz Obrázek 91).



Obrázek 91-Mini úkol ke kytičce

Mašle

Na mašli si ukážeme vnořování cyklů, což znamená, že budeme mít jeden cyklus **for** jako součást těla jiného – nadřazeného – cyklu **for**.

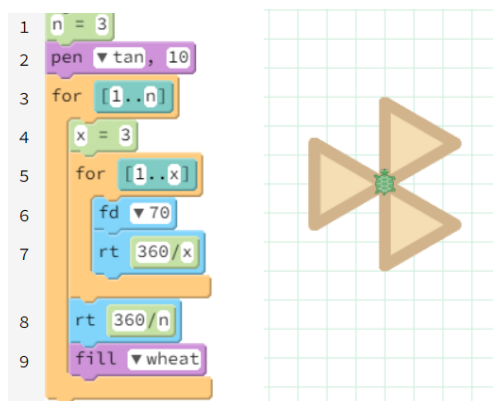


Obrázek 92-Mašle

Náš obrázek mašle se skládá ze tří trojúhelníků. Začneme příkazem pro proměnnou z kategorie **operators**, kterou si pojmenujeme **n**. Proměnná bude značit počet trojúhelníků. Následuje příkaz pro kreslení perem (**pen**). Následuje příkaz pro vytvoření další proměnné, kterou si pojmenujeme **x**. Proměnná bude sloužit pro tvorbu jednoho trojúhelníka.

Poté následují dva cykly. Hlavní cyklus se bude opakovat **n**-krát, protože želvička se otočí dokola a během toho nakreslí tři trojúhelníky. Vnořený cyklus se bude opakovat **x**-krát, protože želvička nakreslí právě jeden trojúhelník o třech stranách a třech úhlech.

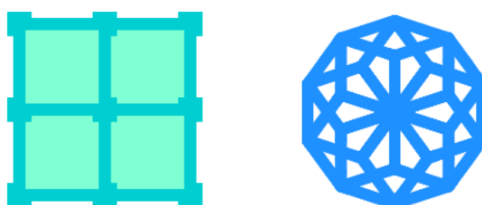
Za vnořeným cyklem bude následovat příkaz pro zatočení (**rt**). Želvička se otočí o úhel tak, aby nám opět koukala nahoru, tudíž zatočí o $360/n$ stupňů. Nakonec vše vybarvíme příkazem **fill**.



Obrázek 93-Hotové řešení

Mini úkol: Žáci si pozmění náš vytvořený úkol, tak aby dosáhli tvaru okna. Okno se skládá ze čtyř čtverců, které jsou vybarvené (**fill**) a ve vrcholech jednotlivých čtverců se nachází čtverečky (**box**). Pozmění si proměnnou **n** na čtyři opakování a proměnnou **x** opět na čtyři, jako čtyři strany čtverce. Do vnořeného cyklu, který nám vykreslí čtverce, přidají nakonec příkaz pro vytvoření čtverečku (**box**).

Případně si vytvoří obrazec, který se skládá z deseti pentagonů. Žáci si do proměnné **n** uloží hodnotu deset, jako deset opakování cyklu. Proměnnou **x** pozmění na číslo pět (jako pentagon o pěti stranách a pěti úhlech).



Obrázek 94-Mini úkol k vnořeným cyklům

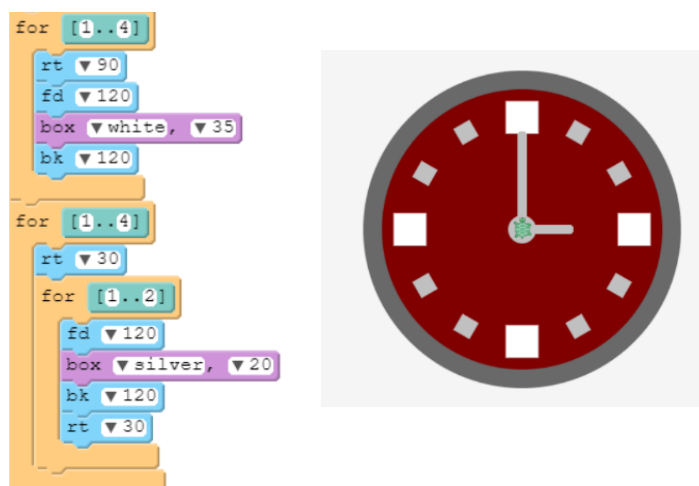
4.17 Doplnková cvičení (k lekci 7)

Doplnková cvičení budou sloužit jako procvičení na doma nebo pro šikovnější žáky jako další cvičení v hodinách. Mají prověřit zvládnutí další lekce ke stávajícím základům. Cvičení jsou náročnější a vyžadují více času a přemýšlení o úlohách. Odkazy k doplnkovým cvičením 4: <http://gawl.pencilcode.net/edit/DOP4/>

Hodiny

Pro procvičení úhlů a složitějších konstrukcí cyklů si žáci nakreslí hodiny. Hodiny jsou nakreslené pomocí příkazů **dot** a **box**. Začneme příkazem **dot** pro vytvoření okraje a pozadí ciferníku. Poté příkazem **box** nakreslíme ciferník tvořený dvanácti kostičkami. Nejdříve nakreslíme čtyři hlavní kostičky, které budou větší a bílou barvou. Mezi každé dvě velké bílé kostičky budou nakreslené dvě menší šedé. Ručičky jsou nakreslené pomocí příkazu **pen** a **fd**. Pro lepší představu je přiloženo řešení cyklů zadaného úkolu (viz Obrázek 95).

Poznámka: Žáci si musí propočítat úhel, o který se želvička bude otáčet mezi jednotlivými kostičkami (tedy 360 stupňů děleno 12 hodinami). To znamená, že se želvička bude otáčet doprava (**rt**) o 30 stupňů.



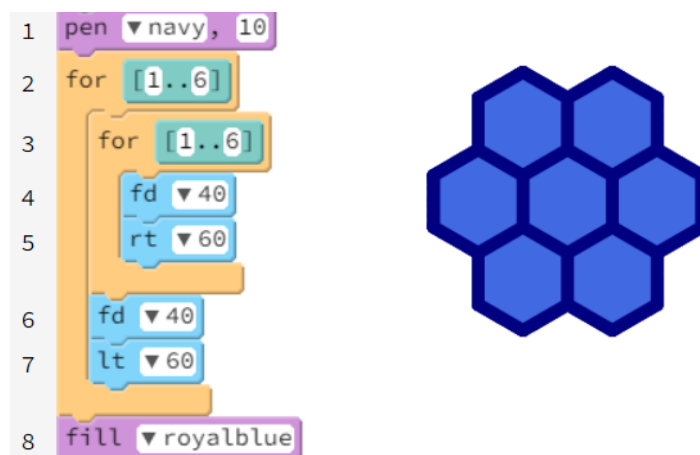
Obrázek 95-Hodiny s ručičkami

AZ-kvív

Obrázek AZ-kvív nakreslíme pomocí dvou cyklů **for**. Začneme příkazem pro kreslení perem (**pen**). Poté následují dva cykly. Hlavní cyklus se bude opakovat šestkrát, protože želvička se otočí dokola a během toho nakreslí šest hexagonů. Vnořený cyklus se bude opakovat také šestkrát, protože želvička nakreslí právě jeden šestiúhelník o šesti stranách a šesti úhlech. Za vnořeným cyklem následují příkazy pro pohyb vpřed (**fd**) a otočení doleva

(**lt**), aby želvička začínala na vrcholu vytvořeného šestiúhelníku. Nakonec vše vybarvíme (**fill**).

Pro lepší představu je přiloženo k obrázku i řešení vnořených cyklů (viz levá část Obrázek 96).



Obrázek 96-AZ-kvíz

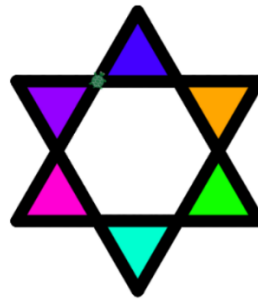
Davidova hvězda

Hvězdu vytvoříme pomocí dvou cyklů **for**. Začneme příkazem pro kreslení perem (**pen**). Dále potřebujeme želvičku vytočit (**rt**) o 30 stupňů, aby nám nakreslila trojúhelník s vodorovnou základnou. Poté následují dva cykly. Hlavní cyklus se bude opakovat šestkrát, protože želvička se otočí dokola a během toho vykreslí šest trojúhelníků. Vnořený cyklus se bude opakovat třikrát, protože želvička nakreslí právě jeden pravidelný trojúhelník. Za vnořeným cyklem následují příkazy pro zatočení (**rt**) a pohyb vpřed (**fd**), aby želvička začínala vždy na vrcholu vytvořeného trojúhelníku. Nakonec následuje příkaz pro vybarvení (**fill**) jednotlivých trojúhelníků.

Žáci si mohou vyzkoušet příkaz **random** z kategorie **operators**, který slouží pro generování náhodných hodnot. Nadefinují mu parametr **color** a uvidí, že každý nakreslený trojúhelník bude vybarven náhodnou barvou.

Pro lepší představu je přiloženo k obrázku i řešení vnořených cyklů (viz Obrázek 97).

```
1 rt 30
2 pen black, 15
3 for [1..6]
4   for [1..3]
5     fd 100
6     rt 120
7   rt 60
8   fd 100
9   fill random color
```



Obrázek 97-Davidova hvězda

5 Ověření

5.1 Ověření daných lekcí

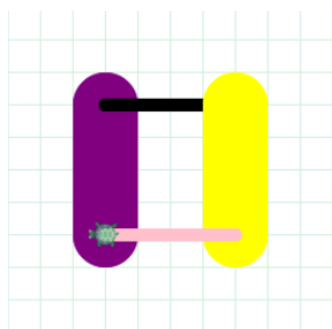
Z důvodu menší časové dotace se podařilo ověřit první čtyři lekce. Testované lekce zahrnují základní práci s příkazy pro pohyb, otáčení, kreslení, vytváření teček a kostiček, práci s úhlem, přesun po souřadnicích a základní práci s cykly. Ověření probíhalo na základní škole Nad Přehradou v Praze 10 v 5. třídě, a to tři vyučovací hodiny. Ověřování se zúčastnilo 22 žáků. Většina žáků byla ve dvojicích, ostatní pracovali samostatně. Každou vyučovací hodinu jsem zahájila výkladem, kde byli žáci seznámeni s příkazy v probírané lekci. Poté jsme si společně zkoušeli příklady a pak jsem nechala žáky pracovat samostatně na jednotlivých cvičeních. Pro šikovnější žáky jsem měla připravený tzv. mini úkol.

Do samostatné práce žáků jsem se snažila zasahovat co nejméně. Chtěla jsem ověřit, zda mnou provedený výklad a příklady v lekcích jsou dostačující pro vypracování konkrétního cvičení. Občas bylo potřeba žákům trochu pomoci zopakováním zadání, nebo zadání podrobněji vysvětlit. Nikdy jsem nezasahovala přímo do jejich práce. Odkazy na výstupy některých žáků jsem uvedla v příloze (viz Příloha B).

Pro lepší orientaci v prostředí a následnou představu, co vše lze tvořit v aplikaci, jsem žákům připravila pracovní list (viz Příloha D). Pracovní list zahrnoval: vysvětlení prostředí, objasnění základních příkazů, zopakování úhlů základních geometrických obrazců a v neposlední řadě i několik obrázkových ukázek z prvních dvou lekcí. Každý z žáků dostal svůj pracovní list, kde si poznamenal své přihlašovací údaje. Počítala jsem s tím, že většina žáků může své přihlašovací údaje zapomenout do doby našeho dalšího setkání. To se také stalo. Díky mé prozíravosti jsme v další vyučovací hodině s žáky mohli hned navázat na hodinu minulou a nemuseli se zdržovat zakládáním nových účtů.

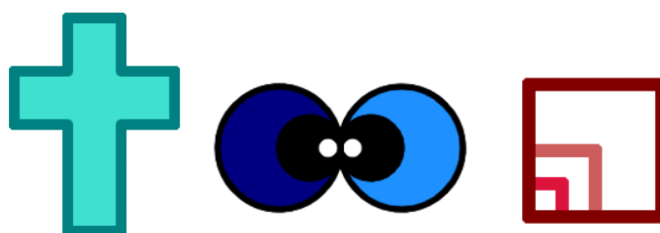
Lekce 1

V první lekci se žáci seznamují s prostředím Pencil Code. Nejdříve dostali pár minut k prohlédnutí prostředí, aby se v něm zorientovali, a následně jsme začali s jednotlivými úkoly. Vyučovací hodinu jsem zahájila vysvětlením tvorby čtverce. Žáci si vyzkoušeli změnit barvu každé strany čtverce a tloušťku obrysu každé strany (viz Obrázek 98). Následně si žáci vyzkoušeli vytvořený čtverec předělat na obdélník.



Obrázek 98-Čtverec s různou barvou strany a tloušťkou obrysu

Dále jsem vysvětlila tvorbu kružnice a trojúhelníka s přidáním příkazem pro vybarvení daného obrazce. Žáci byli velice bystří, proto jsem se rozhodla přejít rovnou na tvorbu kružnice bez využití úlohy „zaoblený čtverec“. Žáci poté tvořili samostatně úlohy: vybarvený kříž, tři čtverce a oči.



Obrázek 99-Samostatné úlohy v první lekci

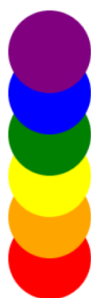
Řešení úloh se žákům dařilo bez větších problémů. Všichni žáci zvládli vypracovat zadané úlohy. Žáci také sami přišli na to, jak odebrat pero, kdy nadeřinovali hodnotu tloušťky pera 0. Tudíž jsem je hned nasměrovala k příkazům na odebrání pera (**pu**) a zapnutí pera (**pd**) z kategorie **art**. Každý žák si tak mohl vybrat způsob, který mu vyhovoval ke splnění úlohy „dva vzdálené čtverce“.



Obrázek 100-Úloha dva vzdálené čtverce

Lekce 2

V druhé lekci se žáci seznámili s tvorbou teček a kostiček. Začala jsem vysvětlením tvorby čtverce s tečkami. Pak žáci dostali za úkol vytvořit duhu z teček nebo kostiček. Záleželo na jejich samostatném výběru.



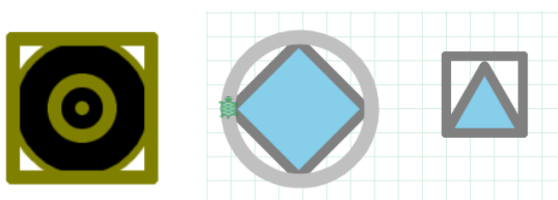
Obrázek 101-Úkol ke druhé lekci: duha

Dále žáci měli nakreslit terč nebo vlajku samostatně jen podle předlohy na interaktivní tabuli. Všem žáků se podařilo úkol splnit.



Obrázek 102-Terč a vlajka k procvičení

Poté jsem žákům vysvětlila tvorbu trojúhelníka v kruhu. Následně se žáci pokusili o zvládnutí kruhu s tečkami či mini úkolu (viz Obrázek 103).



Obrázek 103-Samostatná úloha a mini úkol k trojúhelníku v kruhu

I tento úkol žáci stihli velmi rychle, protože se dobře orientují v úhlech a otáčení želvičky. Žáci byli velmi zvědaví a zkoušeli další a další příkazy v nabídce. Změnili například podobu želvy, načítali obrázky na pozadí, nebo želvu úplně skryli a důvtipně se mě pak dotazovali, „kde ta želva asi je?“

Lekce 3

Ve třetí lekci jsem žákům ukázala příkazy pro přesun želvy po souřadnicích či zrychlení jejího pohybu. Příkazy byly vysvětleny na úloze beruška, kdy měli za úkol doplnit berušce tečky (viz Obrázek 104). Pomalejším žákům byl přesun po souřadnicích znázorněn na tabuli a po mém podrobnějším výkladu to žáci pochopili a byli schopni samostatně pracovat na zadaných úlohách.



Obrázek 104-Beruška

Samostatný úkol si žáci sami mohli vybrat. Rozhodnout se mohli, zdali nakreslí opici nebo „smajlík“. Všem žákům se řešení velmi podařilo.



Obrázek 105-Samostatný úkol smajlík a opice

Další zadání z doplňkových úkolů (sněhulák, mimoň a pavouk) jsem však nezadávala, protože by to žáci z časových důvodů nestihli dokončit. Přešli jsme tedy rovnou na lekci čtvrtou.

Lekce 4

Ve čtvrté lekci se žáci seznámili s pojmem cyklus, který slouží k opakování příkazů. Pochopili ho velmi rychle na úloze čtverec. V našem programu čtverec si poté žáci zkusili změnit parametry čtverce na trojúhelník a vytvořit dva trojúhelníky v cyklu (viz Obrázek 106).



Obrázek 106-Samostatná úloha ke čtvrté lekci

Dále jsem žákům ukázala úlohu schody, načež si žáci založili nový program a vytvořili pilu. Na tabuli jsem vysvětlila celé třídě zadání úlohy, tak aby žáci měli lepší představu o úhlech zatočení želvičky. Barevnost a počet zubů jsem nechala na nich.



Obrázek 107-Samostatná úloha: pila

Ze čtvrté lekce jsem nestihla ověřit proměnné, kvůli malé časové dotaci. Žáci velmi dobře pochopili princip a práci s cykly. Tomuto zjištění přisuzuji velký význam v mém ověřování aplikace Pencil Code na základní škole. Domnívala jsem se, že s cykly budou mít žáci problémy.

5.2 Zhodnocení ověření lekcí

Na základě pozorování během ověřování čtyř lekcí zhodnotím tyto lekce v následujících aspektech: soustředěnost (pozornost) žáků, zájem o činnost v prostředí, vytrvalost (zejména u samostatné práce), emocionální stavy, samostatnost a úspěšnost. Pro tyto aspekty jsem si na každou vyučovací hodinu připravila pozorovací arch, do kterého jsem rovnou zapisovala.

Soustředěnost

První zkušební hodina se uskutečnila ve středu 18. května 2022 od 10:00 do 10:45 na ZŠ Nad Přehradou, Praha 10. Žáci se během hodiny seznamovali s aplikací Pencil Code a jejími základními příkazy. Soustředěnost žáků byla na vysoké úrovni.

Druhá zkušební hodina proběhla v pondělí 23. května od 10:55 do 11:40. Pozornost žáků byla stále na vysoké úrovni.

Třetí zkušební hodina se konala téhož dne v čase 11:50 až 12:35. V této hodině pozornost mírně klesala. Domnívám se, že pokles soustředěnosti žáků mohl být způsobený tím, že se jednalo o druhou hodinu stejného předmětu ve stejný den.

Zájem o činnost v prostředí

Už po první hodině bylo na žácích vidět obrovské nadšení a zájem o prostředí Pencil Code. Žáci se těšili na další vyučovací hodiny a nové dovednosti a poznatky, co si z nich odnesou. Žáci poctivě plnili úlohy a nedělali jiné nepovolené činnosti na internetu.

Dokonce se pár žáků po hodině zeptalo, zda lze Pencil Code stáhnout na mobil či tablet, aby mohli tvořit doma. Což mi přišlo milé, jelikož dnešní děti vyrůstají v digitálním světě plných jiných her a prostředků pro ně atraktivnějších než Pencil Code.

Po skončení ověřování jsem učiněna zpětnou vazbu s žáky, ze které bych ráda vyzdvihla dva poznatky. Zaprvé žáci zmiňovali, že je Pencil Code něco jiného a nového. Není to něco tak běžného, co dělají v jiných hodinách. Zadruhé si přáli více takových hodin a více tvoření v tomto prostředí.

Vytrvalost

Vytrvalost bych zhodnotila u samostatné práce žáků a podle složitosti úloh. Jak již bylo zmíněno výše, žáky samostatná tvorba bavila, proto nedělali žádné nepovolené činnosti. Samostatnou práci se snažili splnit, co nejlepším způsobem. Do úloh byli ponořeni a věnovali jim čas i úsilí.

U náročnějších úloh žáci požádali o pomoc. Proto bylo potřeba tyto žáky nasměrovat. Po upřesnění informací již dále pokračovali v práci sami.

Emocionální stav

Zde bych pozitivně hodnotila to, že žáci nebyli smutní, pokud se jim danou úlohu nepodařilo vytvořit. Rovněž se žádný žák nevztekal nebo nenastala situace, že by někdo zcela rezignoval na vypracování úloh. Pouze jedna žákyně byla smutná, protože seděla ve dvojici se spolužačkou, která ji k vytváření úloh vůbec nepustila. Reagovala jsem poskytnutím notebooku a nového účtu, aby žákyně mohla pracovat samostatně. Z pozdějšího rozhovoru s třídní učitelkou a vedením školy, jsem se dozvěděla, že všichni žáci byli nadšení.

Samostatnost

Při výkladu žáci pracovali zároveň se mnou dle mých instrukcí. Při řešení jednotlivých cvičení pracovali zcela samostatně. Během výuky jsem občas musela pomoci žákům s prací v prostředí, neboť se některým podařilo přepnout do režimu **text code**, či se jim ztratila nabídka příkazů a daných kategorií.

U většiny úloh žáci nepotřebovali pomoc, protože se dobře orientovali v problematice geometrie a úhlech. Nejvíce otázek ze strany žáků přicházelo ve třetí lekci (na téma posunu po souřadnicích). Po znázornění na tabuli a mém podrobnějším výkladu to pomalejší žáci pochopili a byli schopni samostatně pracovat na zadaných úlohách.

Úspěšnost

Jednotlivé úlohy v lekcích byly pro žáky srozumitelné. Úspěšnost plnění samostatných úkolů byla vysoká. Všem žákům se podařilo dojít k požadovanému výsledku. Někteří žáci stihli řešení úlohy v kratším čase. Ostatní žáci potřebovali k vyřešení úloh více času. Všichni žáci však úspěšně vypracovali všechny mnou zadané úlohy. Někteří zvědaví žáci si zkoušeli i další příkazy, které zatím nebyly vysvětleny.

Úpravy lekcí na základě ověřování

Po ověření čtyř lekcí jsem se rozhodla pozměnit lekci 3 a doplňková cvičení k lekcím 1. a 2.

Ze třetí lekce jsem odstranila cvičení: Sluneční soustava z důvodů vysoké časové náročnosti a bezvýznamnosti v oblasti algoritmizace a tvořivosti. Do třetí lekce jsem přesunula dvě úlohy pro procvičení přemísťování se po souřadnicích. Jedná se o úlohy z doplňkových cvičení pro 1. a 2. lekci (opičák a smajlík). Důvod je prostý. Žáci si mohli vybrat ze dvou úloh a pochopili princip přemísťování lépe a rychleji.

Do doplňkových cvičení pro lekce 1. a 2. jsem přidala místo dvou odebraných úloh další dvě úlohy. Chci žákům nabídnout více možností na výběr pro samostatné procvičování. Jedná se o úlohy PokeBall a Peace.

Původní verze před úpravami se nachází v kapitole 4 - praktická část. Úpravy zmiňovaných lekcí jsou v příloze (viz Příloha E). Finální verze lekcí je dostupná v souboru PDF z indexu: <http://gawl.pencilcode.net/>

6 Závěr

Cílem práce bylo analyzovat možnosti aplikace Pencil Code (<http://pencilcode.net/>) pro rozvoj algoritmického myšlení a vytvořit ucelený soubor lekcí obsahující úlohy, které na sebe navazují. Vznikla sada lekcí (<http://gawl.pencilcode.net/edit/>) určená pro rozvoj algoritmického myšlení a výuku programování na základní škole.

Sada mnou navržených úloh je rozdělena do 7 lekcí. Každá lekce obsahuje jak část pro výklad, tak část pro procvičení a samostatnou práci. Jednotlivé lekce taktéž obsahují doplňková cvičení a mini úkoly pro šikovnější a rychlejší žáky, aby se tito žáci nenudili. Zároveň všechny lekce obsahují správné řešení a jsou dostupné online.

Soubor lekcí je určen pro učitele, kterým usnadní výuku programování v prostředí Pencil Code. Soubor obsahuje nabídku více úloh, jak pro výklad, tak pro cvičení. Učitelé si sami zvolí, dle tempa žáků a časové dotace hodin, jaké úlohy a úkoly vykonají či nikoliv.

V průběhu tvoření mých didaktických materiálů v aplikaci Pencil Code jsem jednotlivé úlohy testovala i mimo zmiňovanou ZŠ. Ověření proběhlo na mé sestře (viz Příloha C), která je žákyní 3. třídy základní školy. Díky mému individuálnímu přístupu a vysvětlování úlohy, musím podotknout, že s orientací v aplikaci, přemísťováním se po souřadnicích, programováním, a dokonce i s algoritmickými konstrukcemi (cykly) nebyly žádné problémy. Bohužel ve 3. třídě na základní škole žáci neznají dobře úhly. Z tohoto důvodu se v náročnějších úlohách nedařilo správně užívat příkazy pro otáčení želvičky o žádaný úhel.

První čtyři lekce byly ověřeny na základní škole v 5. třídě. Žáci řešení jednotlivých úloh zvládli zcela samostatně a velmi rychle. Více než polovina žáků plnila mini úkoly nebo se pokoušela o doplňkové úlohy.

Na základě ověřování Pencil Code ve výuce na základní škole jsem dospěla k závěru, že se toto online vývojové prostředí hodí pro rozvoj algoritmického myšlení a výuku programování.

Vzhledem k zvyšující se obtížnosti na sebe navazujících úloh, musí žáci využívat získané dovednosti a znalosti z úloh předchozích a zároveň vstřebávat nové poznatky. Při plnění samostatných úloh a mini úkolů nebylo téměř třeba zasahovat do práce žáků.

Na základně pozorování je nutno dodat, že zájem ze strany dětí o činnost v prostředí byl velký. Žáci se v prostředí Pencil Code rychle zorientovali a pracovali zcela samostatně. Úspěšně splnili zadané úlohy k vypracování. Ráda bych, v rámci příloh, uvedla pár zajímavých výstupů žáků a odkazy na ně (viz Příloha B). Velmi mě potěšilo, když jsem při závěrečné analýze dat zjistila, že si žáci v Pencil Code kreslili sami i dávno po ukončení ověřování.

Vzhledem ke zjištěním při ověřování úloh a samotného programovacího prostředí usuzuji, že aplikace Pencil Code a navržené lekce jsou adekvátní kombinací pro výuku programování na základní škole. Dále mohu konstatovat, že vytyčené cíle této bakalářské práce byly splněny. Zároveň může tato práce sloužit jako návod pro případné navazující lekce v prostředí Pencil Code.

7 Seznam použitých informačních zdrojů

DOHNAL, P. (2008). *Programování a programovací jazyky ve vzdělávání na ZŠ* [online]. Brno, 2009 [cit. 2022-06-21]. Dostupné z: http://is.muni.cz/th/72886/pedf_m/.

FUTSCHEK, Gerald (2006). *Algorithmic Thinking: The Key for Understanding Computer Science*. In. Lecture Notes in Computer Science 4226, Springer

HAVELKOVÁ, H. (2008). *Algoritmy* [online]. Poslední aktualizace 2008-09-30, [cit. 2022-06-21]. Dostupné z: <http://wvc.pf.jcu.cz/ki/data/files/93prg1.ppt>

Pencil Code [online]. Dostupné z: <https://pencilcode.net/#about>

Pencil Code [online]. Dostupné z: <https://pencilcode.net/team.html>

PITNER, T. (2000). Výuka programování na základní a střední škole. [online]. 13-3-2000, [cit. 2022-6-24]. Dostupný z: http://www.fi.muni.cz/~tomp/semuc/text_pitner.html

RVP ZV (2021) *Rámcový vzdělávací program pro základní vzdělávání*– edu.cz. edu.cz – Jednotný metodický portál MŠMT [online]. 2020 [cit. 2022-06-22]. Dostupné z: <https://www.edu.cz/rvp-ramcove-vzdelavaci-programy/ramcove-vzdelavacici-program-pro-zakladni-vzdelavani-rvp-zv/>

ŠTÍPEK, Jiří, 2020. Algoritmizace a programování. In: . PEDF UK.

WIRTH, Niklaus. *Algoritmy a struktury údajov. 2.vyd. Bratislava: Alfa, 1989. ISBN 80-05-00153-3.*

8 Seznam obrázků

Obrázek 1-Příklad zápisu algoritmu (zdroj autorka)	12
Obrázek 2-Algorithmické myšlení (zdroj autorka)	13
Obrázek 3-Prostředí Pencil Code	19
Obrázek 4-Hlavní lišta Pencil Code	19
Obrázek 5-Screenshot ukázky registrace.....	20
Obrázek 6-Odkoušení příkazů pro pohyb a zatáčení želvy a nastavení pera.....	23
Obrázek 7-Čtverec	24
Obrázek 8-Nastavení pera a první strana čtverce	24
Obrázek 9-Hotové řešení	25
Obrázek 10-Mini úkol ke čtverci.....	25
Obrázek 11-Zaoblený čtverec.....	25
Obrázek 12-Hotové řešení	26
Obrázek 13-Mini úkol k zaoblenému čtverci	26
Obrázek 14-Kružnice.....	27
Obrázek 15-Hotové řešení	27
Obrázek 16-Mini úkol ke kružnici	28
Obrázek 17-Rovnostranný trojúhelník	28
Obrázek 18-Obrázek k úhlům trojúhelníka	29
Obrázek 19-Hotové řešení	29
Obrázek 20-Dva čtverce pod sebou.....	30
Obrázek 21-Kopírování příkazů	30
Obrázek 22-Hotové řešení	31
Obrázek 23-Mini úkol ke dvěma čtvercům	31
Obrázek 24-Druhý mini úkol ke dvěma čtvercům	32
Obrázek 25-Tři čtverce v sobě.....	32
Obrázek 26-Kříž	33
Obrázek 27-Očička	33
Obrázek 28-Domeček jedním tahem	34
Obrázek 29-Tečky různými barvami	34

Obrázek 30-Hotové řešení	35
Obrázek 31-Mini úkol k tečkám	35
Obrázek 32- „Tečko-čtverec“	36
Obrázek 33-Hotové řešení	36
Obrázek 34-Mini úkol k "tečko-čtverci"	37
Obrázek 35-Překreslování příkazů	37
Obrázek 36-Terč	37
Obrázek 37-Hotové řešení	38
Obrázek 38-Mini úkol k terči	39
Obrázek 39-Vlajka.....	39
Obrázek 40-Hotové řešení	40
Obrázek 41-Mini úkol ke vlaječce	40
Obrázek 42-Trojúhelník uvnitř kruhu.....	41
Obrázek 43-Hotové řešení	41
Obrázek 44-Mini úkol k trojúhelníku uvnitř kruhu	42
Obrázek 45-Duha z teček	42
Obrázek 46-Kruh ve čtverci s tečkami	43
Obrázek 47-Kapka s bambulí	43
Obrázek 48-Semafor.....	44
Obrázek 49-Smajlík.....	44
Obrázek 50-Lod'ka jedním tahem.....	45
Obrázek 51-Opičák.....	45
Obrázek 52-YinYang.....	45
Obrázek 53-BMW	46
Obrázek 54-Beruška	47
Obrázek 55-Příklad hotového řešení	48
Obrázek 56-Mimoň	49
Obrázek 57-Sněhulák	49
Obrázek 58-Pavouk	50
Obrázek 59-Čtverec pomocí cyklu	50
Obrázek 60-Hotové řešení	51

Obrázek 61-Tvary v cyklu.....	51
Obrázek 62-Hotové řešení	52
Obrázek 63-Mini úkol k hadovi.....	52
Obrázek 64-Schody	52
Obrázek 65-Hotové řešení	53
Obrázek 66-Mini úkol ke schodům	53
Obrázek 67-Desetiúhelník	54
Obrázek 68-Hotové řešení	54
Obrázek 69-Mini úkol k desetiúhelníku	55
Obrázek 70-Trojúhelníky	56
Obrázek 71-Pila	56
Obrázek 72-Čtverec s tečkami.....	56
Obrázek 73-Audi	57
Obrázek 74-Vločka.....	57
Obrázek 75-Hvězda	58
Obrázek 76-Kytka.....	58
Obrázek 77-Tvorba funkce Obdelnicek	60
Obrázek 78-Vlajka Polska v prostředí Pencil Code	60
Obrázek 79-Hotové řešení	61
Obrázek 80-Vlajka Česka v prostředí Pencil Code	61
Obrázek 81-Tvorba funkce Trojuhelnicek	62
Obrázek 82-Hotové řešení volání funkcí pro tvorbu vlajky Česka	62
Obrázek 83-Mini úkol k funkcím	63
Obrázek 84-Příklady dopravních značek.....	63
Obrázek 85-Hotové řešení	64
Obrázek 86-Mini úkol ke klávesám	65
Obrázek 87-Cvičení ke klávesám	65
Obrázek 88-Kytička z kružnic	66
Obrázek 89-Hotové řešení cyklus for.....	67
Obrázek 90-Hotové řešení cyklus while.....	67
Obrázek 91-Mini úkol ke kytice	68

Obrázek 92-Mašle.....	68
Obrázek 93-Hotové řešení	69
Obrázek 94-Mini úkol k vnořeným cyklům	69
Obrázek 95-Hodiny s ručičkami.....	70
Obrázek 96-AZ-kvíz.....	71
Obrázek 97-Davidova hvězda	72
Obrázek 98-Čtverec s různou barvou strany a tloušťkou obrysu	74
Obrázek 99-Samostatné úlohy v první lekci.....	74
Obrázek 100-Úloha dva vzdálené čtverce	74
Obrázek 101-Úkol ke druhé lekci: duha.....	75
Obrázek 102-Terč a vlajka k procvičení.....	75
Obrázek 103-Samostatná úloha a mini úkol k trojúhelníku v kruhu.....	75
Obrázek 104-Beruška	76
Obrázek 105-Samostatný úkol smajlík a opice	76
Obrázek 106-Samostatná úloha ke čtvrté lekci	77
Obrázek 107-Samostatná úloha: pila.....	77
Obrázek 108-Peace	VII
Obrázek 109-Lod'ka jedním tahem.....	VII
Obrázek 110-PokeBall.....	VIII
Obrázek 111-YinYang.....	VIII
Obrázek 112-BMW	IX
Obrázek 113-Beruška	X
Obrázek 114-Smajlík.....	X
Obrázek 115-Opičák.....	XI

9 Seznam zkratek

RVP	Rámcově vzdělávací program
ŠVP	Školní vzdělávací program
ZV	Základní vzdělávání
fd	příkaz jít rovně
bk	příkaz jít pozadu
rt	příkaz zatočit doprava
lt	příkaz zatočit doleva
rt,100	příkaz zatočit doprava o poloměr 100 bodů
lt,100	příkaz zatočit doleva o poloměr 100 bodů
pu	příkaz odebrání pera
pd	příkaz navrácení pera
fill	příkaz vybarvení obrazce
pen	příkaz pera
dot	příkaz vytvoření tečky
box	příkaz vytvoření kostičky
jumpto	skok na souřadnice x a y
moveto	přesun na souřadnice x a y
home	počáteční souřadnice a úhel želvičky
speed	zrychlení pohybu želvičky
random	příkaz pro náhodnou proměnnou
button	příkaz pro tlačítko
label	příkaz pro textové pole

10 Seznam příloh


Příloha A-ŠVP ZŠ Nad Přehradou (Matematika a její aplikace 5. třída).....	I
Příloha B-Výstupy dětí	II
Příloha C-Výstup sestry	IV
Příloha D-Pracovní list pro žáky	V
Příloha E-Finální verze lekcí po úpravách	VII

Příloha A-ŠVP ZŠ Nad Přehradou (Matematika a její aplikace 5. třída)

RVP výstupy	Školní výstupy	Učivo	Přesahy a vazby	poznámky
1. tématiký okruh RVP – 2. období				
Číslo a početní operace Zaokrouhluje přirozená čísla, provádí odhady a kontroluje výsledky početních operací v oboru přirozených čísel.	Zapíše a přečte čísla do milionu a přes milion. Zaokrouhluje přirozená čísla. Odhaduje a kontroluje výpočty početních výkonů.	Obor přirozených čísel zaokrouhlování čísel		
Provádí písemné početní operace v oboru přirozených čísel.	Sečítá a odčítá přirozená čísla z paměti i písemně. Pamětně násobí a dělí přirozená čísla. Písemně násobí až trojčlenným činitelem. Písemně dělí jednociferným a dvojciferným dělitelem.	Početní operace s přirozenými čísly sčítání a odčítání z paměti i písemně násobení a dělení písemné algoritmy početních operací	OSV – osobnostní rozvoj, schopnost sebekontroly, rozvoj schopnosti poznávání morální rozvoj-řešení problémů, sebeovládání	
Využívá při pamětném i písemném počítání komutativnost a asociativnost sčítání a násobení.	Využívá vlastnosti početních operací (komutativnost a asociativnost).	Vlastnosti početních operací s přirozenými čísly		
Řeší a tvoří úlohy, ve kterých aplikuje osvojené početní operace v celém oboru přirozených čísel.	Řeší a tvoří jednoduché a složené slovní úlohy, ve kterých aplikuje osvojené početní výkony v oboru přirozených čísel.	Slovní úlohy	MKV – lidské vztahy, základní morální normy OSV – komunikace, mezilidské vztahy, práce ve skupině	
	Zapíše a přečte desetinné číslo řádu desetin a setin. Zaokrouhlí desetinné číslo řádu desetin a setin na celky. Řeší jednoduché slovní úlohy na užití desetinných čísel.	Desetinná čísla zaokrouhlování desetinných čísel slovní úlohy		rozšiřující učivo
2. tématiký okruh RVP – 2. období				
ZÁVISLOSTI, VZTAHY A PRÁCE S DATY Vyhledává, sbírá a třídí data.	Zapíše, použije data z grafu ve čtvercové síti.	Diagramy, grafy, tabulky		
Čte a sestavuje jednoduché tabulky a diagramy.	Čte a sestavuje diagram a jednoduché grafy v soustavě souřadnic. Doplní řady čísel, tabulky.			
3. tématiký okruh RVP – 2. období				
Geometrie v rovině a prostoru Narýsuje a znázorní základní rovinné útvary (čtverec, obdélník, trojúhelník); užívá jednoduché konstrukce.	Uplatňuje pojmy rovina, polovina, trojúhelník pravouhlý, rovnostranný, rovnoúhelníkový. Narýsuje čtverec, obdélník, trojúhelník. Dbá na přesnost a čistotu rýsování.	Základní útvary v rovině čtverec, obdélník, trojúhelník	EV – vztah člověka k prostředí lidské aktivity a problémy životního prostředí	
Určí obsah obrazce pomocí čtvercové sítě a užívá základní jednotky obsahu.	Určí obsah rovinných obrazců pomocí čtvercové sítě. Užívá základní jednotky obsahu mm^2 , cm^2 , m^2 . Řeší slovní úlohy na výpočty obsahů čtverce a obdélníku.	Obsah obrazce jednotky obsahu		
Rozpozná a znázorní ve čtvercové síti jednoduché osově souměrné útvary a určí osu souměrnosti útvaru překládáním papíru.	Pozná a znázorní ve čtvercové síti jednoduchý, osově souměrný útvar. Určí osu souměrnosti útvaru překládáním papíru.	Osově souměrné útvary		
4. tématiký okruh RVP – 2. období				
Nestandardní aplikační úlohy a problémy Řeší jednoduché praktické slovní úlohy a problémy, jejichž řešení je do značné míry nezávislé na obvyklých postupech a algoritmech školské matematiky.	Řeší magické čtverce. Užívá prostorovou představivost.	Magické čtverce Prostorová představivost		

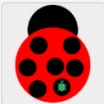
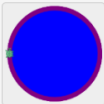





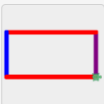





Příloha B-Výstupy dětí

<https://sejracek12.pencilcode.net/edit/>


sejracek12 

directory

Q Filter






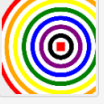


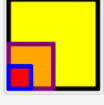


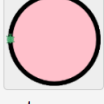

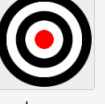
			
beruska	kruznice	trojuhelnik	New file
			
beruskam	oci	untitled	
			
ctverec	tecka	untitled2	
			
emoiim	terc	untitled3	

<https://kocour12.pencilcode.net/edit/>

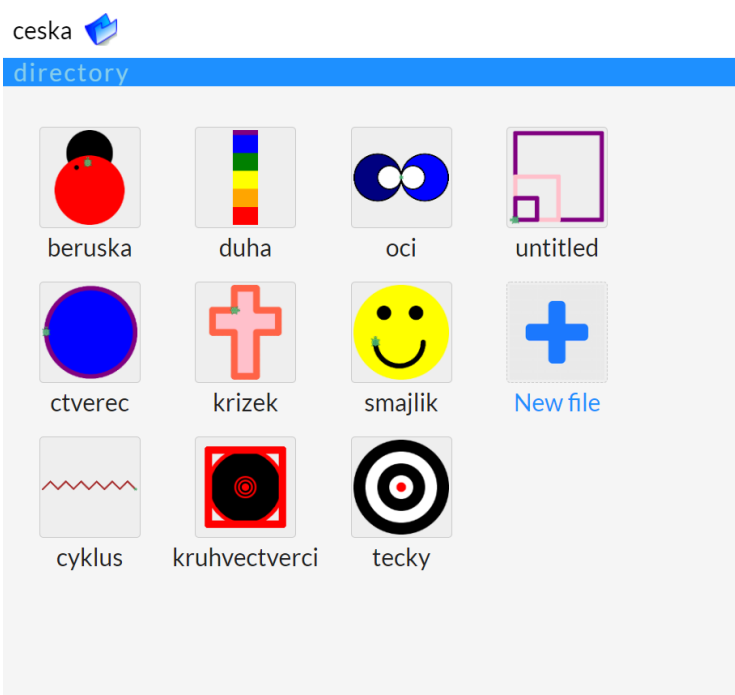
kocour12 

directory

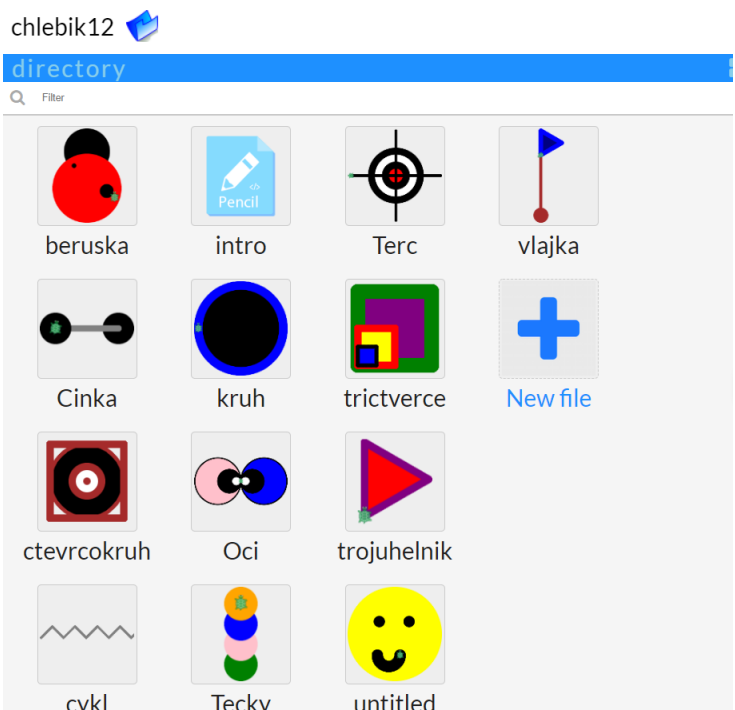
Q Filter

			
beruska	cyklus	smajlik	vlajka
			
BERUSKakk	duha	SMAJLIKkk	New file
			
ctverce	kriz	tecky	
			
ctverec	oci	terc	

<https://ceska.pencilcode.net/edit/>



<https://chlebik12.pencilcode.net/edit/>



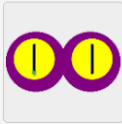
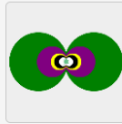
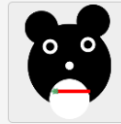

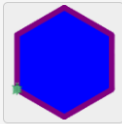

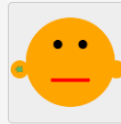
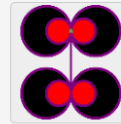




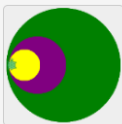
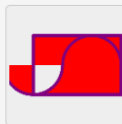

Příloha C-Výstup sestry

<http://gawlow.sk.pencilcode.net/edit/AGATA/>

gawlow.sk  AGATA

directory

Q Filter

			
aga	agatkruhy	medabeda	Test5
			
agatcyklus	agatschodyc...	oblycej	zkouskaa
			
agaticeopice	agattvarycykl	skorokaktus	New file
			
Agatka2	agi	test	

Příloha D-Pracovní list pro žáky

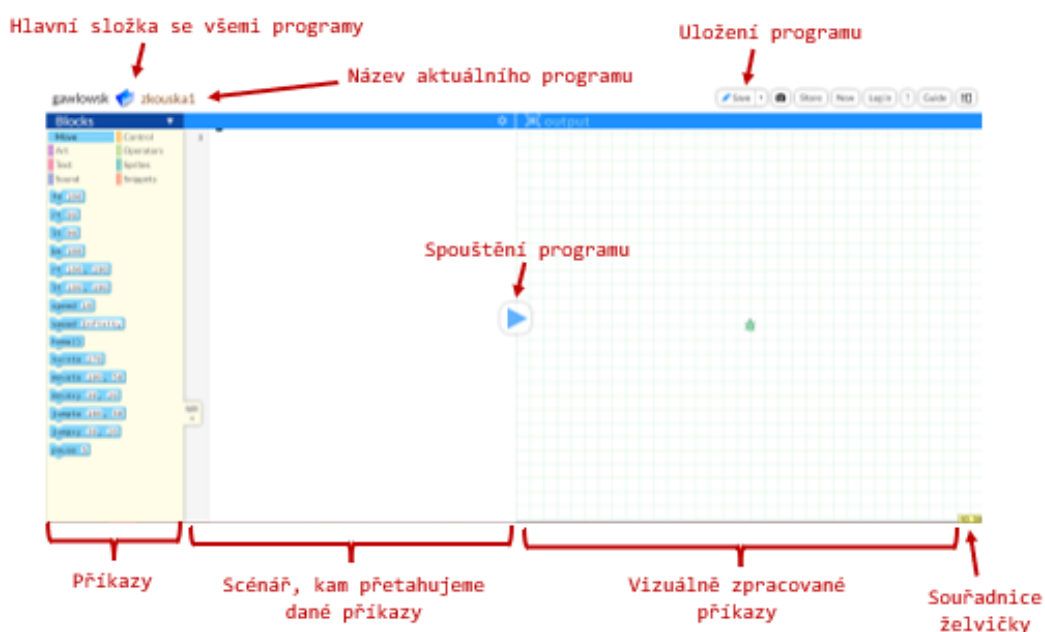
Pracovní list Pencil code

Odkaz: www.pencilcode.net

Přihlašovací údaje:

Jméno:	
Heslo:	

Popis prostředí:



Základní příkazy:

fd-pohyb vpřed

rt 90-zatočení vpravo o 90°

Lt 90-zatočení vlevo o 90°

rt 180,100-zatočení doprava o 180° s poloměrem 100

Lt 180,100-zatočení doleva o 180° s poloměrem 100

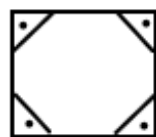
pen-nastavení barvy pera a jeho tloušťky

fill-vybarvení obrazce barvou

Popis základních geometrických konstrukcí:

Čtverec:

-čtyři stejně dlouhé strany a čtyři stejné vnitřní úhly

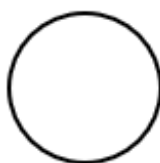


Pravý úhel 90°

Příkazem *rt 90;*

Kružnice:

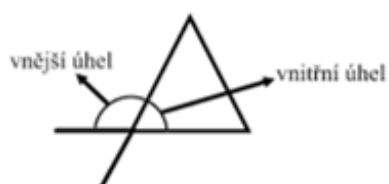
-celkem 360°



Příkazem *rt 360, 100;*

Případně použijeme 4x příkaz *rt 90, 100;*

Trojúhelník:



Pro nakreslení úhlu 60° ,
potřeba vytočit želvičku o
úhel 120° .

Příkazem *rt 120;*

Úlohy k procvičení Základy:



Příloha E-Finální verze lekcí po úpravách

Doplňková cvičení (k lekcím 1 a 2)

Doplňková cvičení budou sloužit jako procvičení na doma nebo pro šikovnější žáky jako další cvičení v hodinách. Mají prověřit zvládnutí lekce: Základy 1 a 2. Cvičení jsou náročnější a vyžadují více času a přemýšlení o úlohách. Odkazy k doplňkovým cvičením 1: <http://gawl.pencilcode.net/edit/DOP1/>

Peace

Znak Peace můžeme vytvořit dvěma způsoby. Prvním způsobem může být nakreslení tečky (**dot**), která bude černou barvou. Následně nakreslíme tečku bílou, která bude o něco menší než černá. Dále použijeme příkazy **fd**, **bk** a **rt**, abychom nakreslili vnitřek znaku.

Druhým způsobem může být nakreslení kružnice a následně její vybarvení příkazem **fill**. Pro nakreslení vnitřku se bude pokračovat stejnými příkazy jako u prvního způsobu.



Obrázek 108-Peace

Lod'ka

Dalším doplňkovým úkolem bude lod'ka jedním tahem. Pomocí příkazu **fd** a úhlem zatočení **rt** s poloměrem otáčení **r**, vytvoříme tělo lodi, které vybarvíme modrou. Následně vytvoříme stožár a vlaječku, kterou vybarvíme červenou barvou.



Obrázek 109-Lod'ka jedním tahem

PokeBall

PokeBall vytvoříme pomocí dvou půlkružnic. První bude nakreslená černou barvou pera (**pen**) a vybarvená červeně příkazem **fill**. Druhá bude nakreslená také černou barvou (**pen**) a vybarvená bílou příkazem **fill**. Následně vytvoříme tučnou čáru a černou a bílou tečkou (**dot**) vnitřek PokeBallu.



Obrázek 110-PokeBall

YinYang

YinYang vytvoříme pomocí dvou půlkružnic (příkazy **rt** o poloměr otáčení) a následně pro ohraničení velkou půlkružnicí (opět příkaz zatočení **rt** s poloměrem otáčení **r**) černou barvou. Následně vyplníme bílou barvou pomocí příkazu **fill**. To samé zopakujeme u spodní černé části. Poté se přemístíme příkazy **fd** a uděláme tečky (**dot**).



Obrázek 111-YinYang

BMW

Další úkol je znak BMW, který nakreslíme pomocí černé a bílé tečky (**dot**). Následně vytvoříme dvě čtvrtky kruhu šedou barvou příkazem **fd** a příkazem zatočení o úhel **rt** s poloměrem **r**. Poté je vybarvíme modrou barvou.



Obrázek 112-BMW

Lekce 3

Vstupní požadavky na žáky:

- Zvládají přetahování příkazů na dané místo ve scénáři a tvoření složitějších obrazců.
- Orientují se ve výstupu programu a zvládají pozměnit program dle dalších instrukcí a různých předloh jiných tvarů a obrazců.
- Pracují se základními příkazy z kategorie **move** (**fd**, **bk**, **rt/lt**, **rt/lt** s poloměrem otáčení **r**)
- Pracují se základními příkazy z kategorie **art** (**dot**, **box**, **pen**, **fill**, **pu** a **pd**).
- Žáci zvládají kopírovat příkazy ve scénáři a vkládat na určené místo.

Nové dovednosti:

- Osvojení si příkazů **jumpto**, **moveto** a **home** z kategorie **move** pro přemístění po souřadnicích.
- Žáci se naučí nový příkaz **speed** z kategorie **move** pro zrychlení pohybu želvičky ve výstupu.

Výklad k souřadnicím

Želvička se umí přemísťovat po souřadnicích pomocí kurzoru myši a příkazům k tomu určeným. Na začátku je nastavená na souřadnicích [0:0]. Příkazem **jumpto** želvička skočí na určitou souřadnici bez toho, aby zanechala stopu pera. Příkazem **moveto** se želvička posune přímým směrem na určitou souřadnici s tím, že zanechá stopu pera. Příkazem **home** se želvička navrátí na výchozí pozici a to souřadnice [0:0]. Příkaz **home** také navrací želvičku při otáčení do výchozí pozice 0° (bude vždy zpátky koukat nahoru).

Jako příklad pro vysvětlení jsem zvolila Berušku. Nakreslíme si s dětmi černou tečku (**dot**), která bude značit hlavu a poté větší červenou tečku (**dot**) pro tělo berušky. Poté dětem ukáží příkaz **jump to**, díky kterému dokreslíme tečky různé velikosti a na různé místa v těle berušky (viz Obrázek 54).



Obrázek 113-Beruška

Cvičení k souřadnicím

Smajlík

Smajlíka vytvoříme pomocí jedné větší tečky. Následně využijeme příkazy **fd** a **bk**, kterými se budeme přemísťovat, abychom nakreslili oči a ústa (**dot**, úhel zatočení **rt** a poloměr otáčení **r**).



Obrázek 114-Smajlík

Opičák

Další úkol bude vytvořit opičku. Začneme příkazem **dot** pro vytvoření ouška. Přidáme mezeru pomocí **fd** a následně vytvoříme další ouško. Přesuneme se doprostřed mezi ouška

a pomocí tečky nakreslíme hlavu. Příkazy **fd** a **bk** (pro přesouvání), **dot** a úhel zatočení **rt** s poloměrem otáčení **r** dokreslíme opičákovi oči a pusu.



Obrázek 115-Opičák