



Pedagogická
fakulta
Faculty
of Education

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

Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta
Katedra informatiky

Diplomová práce

Výuka základů programování v prostředí Scratch

Vypracoval: Jan Krejsa
Vedoucí práce: doc. PaedDr. Jiří Vaníček, Ph.D.

České Budějovice 2014

Název práce: Výuka základů programování v prostředí Scratch

Autor: Jan Krejsa

Katedra: Katedra Informatiky

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

E-mail vedoucího: vanicek@pf.jcu.cz

ANOTACE

Práce se zabývá výukou programování na základních školách v programu Scratch. Teoretická část obsahuje průzkum, který byl proveden pomocí dotazníku na základních školách a který ukazuje skutečnosti týkající se výuky programování. Dále je Scratch porovnán s programy Baltík a Imagine Logo, které se nejvíce používají na základních školách pro výuku programování.

Práce také obsahuje analýzu materiálů dostupných pro výuku Scratch. Výsledky analýzy výukových materiálů jsou graficky zobrazeny v přehledném grafu. Na základě této analýzy je vytvořen postup výuky, který je rozdělen do dvanácti témat. Pro každé téma je vytvořen žákovský list a metodické pokyny pro učitele.

KLÍČOVÁ SLOVA

Algoritmizace, programování, Scratch, základní škola, výuka

Title: Education of basic programming in Scratch environment

Author: Jan Krejsa

Department: Katedra Informatiky

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

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

ANOTATION:

The work deals with programming education on primary schools in Scratch programme. Investigation was carried out by questionnaire on primary school in theoretic part. The investigation shows the facts relating to teaching programming. Scratch is compared with Baltík and Imagine Logo, which are the most commonly used on primary schools.

The work also includes an analysis of teaching materials that are available for teaching Scratch. Results of the analysis of teaching materials are shown in a graph. Procedure of teaching was made on the basis of this analysis. The procedure of teaching is divided into twelve subjects. Pupil sheet and guideline for teacher is created for each subject.

KEYWORDS:

Algorithm, development, programming, Scratch, primary school, education

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

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

30. 4. 2014

Poděkování

Děkuji doc. PaedDr. Jiřímu Vaničkovi, Ph. D. za informace, trpělivost, rady a materiály, které mi poskytl během vypracovávání diplomové práce.

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jan KREJSA**
Osobní číslo: **P090120**
Studijní program: **N7503 Učitelství pro základní školy**
Studijní obory: **Učitelství fyziky pro 2. stupeň základních škol**
Učitelství výpočetní techniky pro 2. stupeň základních škol
společný základ - navazující
Název tématu: **Výuka základů programování v prostředí Scratch**
Zadávající katedra: **Katedra informatiky**

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

Scratch je jeden z vedoucích světových produktů určených pro výuku základů programování. Je zdarma a dostupný v české verzi. Je vyvíjen na Massachusetts Institute of Technology. Úkolem studenta je zmapovat možnosti výuky základů programování ve vývojovém didaktickém prostředí Scratch. Porovná toto prostředí s dalším podobným nástrojem, speciálně s Imagine LOGO. Student porovná zejména možnosti programu v oblasti výuky, komfort a uživatelskou přívětivost prostředí. Dále porovná snadnost zápisu a editace algoritmů, práce s daty, práce s grafikou a multimédií, práce s objekty. Další kritéria porovnání mohou být intuitivnost prostředí, dostupnost a zpracování dokumentace. Student vytvoří vzorový projekt, v němž popíše specifika programování v daném prostředí. Dále vytvoří sadu výukových programovacích cvičení pro výuku základů programování v tomto prostředí tak, aby podle této pomůcky mohl uživatel vytvářet jednoduché programy, které budou učit základní pojmy. Tuto sadu výukových materiálů ověří v praxi a provede analýzu tohoto ověření.

Rozsah grafických prací: CD ROM
Rozsah pracovní zprávy: 60
Forma zpracování diplomové práce: tištěná

Seznam odborné literatury:

1. RESNICK, Mitchel. Scratch.mit.edu [online].
2. 7. 2009 [cit. 2011-04-21]. Getting Started Guide.
Dostupné z WWW: <http://info.scratch.mit.edu/sites/infocratch.media.mit.edu/files/file/ScratchReferenceGuide14.pdf>
2. RESNICK, Mitchel. Scratch.mit.edu [online]. 13. 5. 2009 [cit. 2011-04-21]. Getting Started Guide.
Dostupné z WWW: <http://info.scratch.mit.edu/sites/infocratch.media.mit.edu/files/file/ScratchGettingStartedv14.pdf>
3. BLAHO, A., KALAŠ, I., Imagine Logo Učebnice programování pro děti. Computer Press, Brno, 2006. ISBN 80-251-1015-X.
4. VANÍČEK, J., MIKEŠ, R., Informatika pro základní školy a víceletá gymnázia 3. Díl. Computer Press, Brno, 2006. ISBN 80-251-1082-6.
5. ALIMISI, R., WINTERS, N. Programming playfully for a real-life problem: conditional statements on the stage of Scratch. In Clayson, J. E., Kalaš I. (eds.) Constructionist approaches to creative learning, thinking and education: lessons for the 21st century, proceedings Constructionism 2010, Paris 16.-20. 8. 2010. Bratislava: Library and publishing centre Comenius University, 2010. ISBN 978-80-89186-65-5.

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

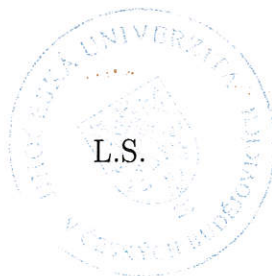
Datum zadání diplomové práce: 12. dubna 2011

Termín odevzdání diplomové práce: 30. dubna 2012



Mgr. Michal Vančura, Ph.D.

děkan



PaedDr. Jiří Vaníček, Ph.D.

vedoucí katedry

V Českých Budějovicích dne 8. dubna 2010

Obsah

1. Úvod	10
1.1 Cíle práce.....	10
1.2 Metody práce	11
2. Průzkum potřebnosti mé práce	13
2.1 Dotazník	13
2.2 Výsledky dotazníkového šetření	13
2.3 Shrnutí výsledků průzkumu.....	17
3. Porovnání programů pro výuku programování na ZŠ.....	18
3.1 Baltík	18
3.2 Imagine Logo	20
3.3 Scratch	21
3.4 Porovnání programů	23
4. Materiály dostupné k výuce programování na ZŠ.....	25
4.1 Tvořivá informatika s Baltíkem	25
4.2 Imagine Logo Učebnice programování pro děti.....	26
4.3 Bakalářská práce Možnosti využití jazyka Scratch na ZŠ	27
4.4 Materiály pro Scratch dostupné na webu	28
4.4.1 Learn Scratch.....	28
4.4.2 Introduction to Programming	29
4.4.3 Scratch Lessons: Shall We Learn Scratch Programming for Tweens..	30
4.4.4 Resources for teachers, parents and computer club facilitators	31
4.4.5 LearnScratch.....	31
4.4.6 Scratch curriculum guide.....	33
4.5 Shrnutí	34
5. Programování ve Scratch 2.0.....	37
5.1 Seznámení s prostředím editoru Scratch	37
5.2 Vzorový projekt.....	39
6. Metodika výuky.....	47

6.1 Osnova výuky	47
6.2 Postup výuky	50
6.2.1 Seznámení s prostředím Scratch.....	51
6.2.2 Posloupnost příkazů a cyklus	53
6.2.3 Postavy a pozadí	56
6.2.4 Více postav a postupný vývoj	61
6.2.5 Paralelizace a události	63
6.2.6 Události a zprávy	66
6.2.7 Testování a ladění.....	70
6.2.8 Klonování	74
6.2.9 Podmíněný příkaz a operátory.....	77
6.2.10 Proměnné.....	80
6.2.11 Bloky (Procedury)	84
6.2.12 Závěrečný projekt.....	87
7. Ověření postupu výuky v praxi	89
7.1 Ověření v rámci zájmového kroužku	89
7.2 Výuka Scratch v rámci projektových dnů	94
8. Závěr.....	96
Literatura a ostatní použité zdroje	98
Přílohy	100

1. Úvod

Téma diplomové práce Výuka základů programování v prostředí Scratch jsem si vybral, protože jsem chtěl vytvořit materiál, který by podporoval výuku programování. Jsem totiž zastáncem názoru, že by se mělo na základních školách vyučovat programování, protože rozvíjí u žáka především kreativitu, analytické myšlení, inženýrské a matematické dovednosti.

Učitelé na základních školách programování učí spíše jen výjimečně. Vstříc jim nevychází ani nabídka programů, které jsou k dispozici na českém trhu, ani podpora ve formě školení. Mým cílem je změnit pohled učitelů a škol na problematiku výuky programování v hodinách výpočetní techniky. Chtěl bych ukázat, že zavedení programování do výuky může být jednoduché a učitelé ani školy se toho nemusí obávat.

Moderních specializovaných programů, které by sloužili k výuce programování na základních školách, není mnoho. Jedním z nich je program Scratch. Jedná se o jednoduché výukové volně šiřitelné programovací prostředí, vyvinuté na Massachusetts Institute of Technology. Umožňuje programovat za pomoci bloků příkazů, které se do sebe skládají podobně jako puzzle. Je vhodný především pro využití ve škole a při zájmových činnostech dětí, které jsou orientované na práci s počítačem. Dnes je rozšířen po celém světě. Určen je mládeži od šesti do šestnácti let bez předchozích zkušeností s programováním. Lze ho zdarma a neomezeně užívat. Prostor Scratch se neinstaluje, je dostupné jako cloud aplikace na portálu „scratch.mit.edu“.

Velmi mě zajímal současný stav výuky programování v základních školách, proto jsem provedl průzkum, z jehož výsledků má práce vycházet. Zabývám se také dostupnými materiály pro výuku programování, které v případě prostředí Scratch jsou pouze v anglickém jazyce.

Ve své práci navrhuji vlastní postup výuky programování ve Scratch a také soubor úloh. Ke každému tématu mám připraveno až několik úloh, které učitelé mají k dispozici a mohou je použít ve výuce.

1.1 Cíle práce

Průzkum na základních školách

Udělal jsem průzkum na základních školách a zjistil jsem, jestli v rámci informatiky vyučují programování a v jakém programovacím prostředí.

Porovnání programů pro výuku programování na ZŠ

Porovnávám programy Scratch 2.0, Imagine Logo a Baltík. Ve své práci podávám o těchto programech základní informace, včetně možností jejich získání a používání.

Porovnání dostupných výukových materiálů

Podávám základní informace o materiálech pro výuku Scratch, které jsou dostupné na portálu scratched.media.mit.edu. Tyto materiály také porovnávám z hlediska postupu výuky s materiály Imagine Logo a Baltík.

Vzorový projekt

Práce obsahuje vzorový projekt, na kterém ukazují specifika programování v daném prostředí Scratch.

Návrh vlastního postupu výuky

Při návrhu vlastního postupu výuky vycházím z dostupných materiálů, které jsem porovnával a také z materiálů od uznávaných odborníků.

Metodická příručka a soubor žákovských listů

Metodická příručka pro učitele a žákovské listy jsou provázány se studií na portálu Scratch. Studio určené pro učitele obsahuje vypracované projekty, které jsou zadávány žákům a jsou popisovány v metodice.

Ověření navrženého postupu výuky

Ověření navrženého postupu výuky jsem provedl v rámci zájmového kroužku, který mám možnost vést na škole, kde vyučuji.

1.2 Metody práce

Průzkum na základních školách jsem provedl pomocí dotazníkového šetření. Získaná data jsem analyzoval a vyhodnotil s pomocí tabulek a grafů.

V rámci přípravy na diplomovou práci jsem prostudoval dostupné materiály určené k výuce programování v prostředí Scratch. Studoval jsem také další odborné a didaktické materiály, které se vztahují k výuce informatiky a programování na základních školách.

Analyzoval jsem dostupný software pro výuku programování na základních školách. Ve své práci uvádím porovnání programů Scratch 2.0, Imagine Logo a Baltík.

Na základě komparace metodických materiálů dostupných pro prostředí Scratch 2.0 jsem vyvinul vlastní metodiku. Tuto metodiku porovnávám s dostupnými

materiály pro Scratch 2.0 a také s materiály, které jsou k dispozici pro programy Imagine Logo a Baltík. Srovnání jsou v práci přehledně graficky znázorněna.

Spolu s metodickými pokyny pro učitele jsem vytvořil i žákovské listy, které jsou přílohou práce. Pro učitele a žáky jsou k dispozici tzv. studia s projekty, které jsem naprogramoval pro podporu výuky.

Při ověření postupu výuky a výukových materiálů jsem použil metodu zúčastněného pozorování. V průběhu ověření byly analyzovány práce žáků, které vytvořili v prostředí Scratch.

2. Průzkum potřebnosti mé práce

V první fázi své práce jsem provedl průzkum na základních školách. V průzkumu jsem měl v plánu oslovit 2504 základních škol, na které jsem našel emailové adresy.

2.1 Dotazník

Oslovení základních škol jsem provedl formou dotazníku, který mám vytvořený pomocí aplikace „Dokumenty Google“. Informace o sdíleném dotazníku dostali potenciální respondenti v krátkém motivačním emailu. Kontakty jsem čerpal z portálu „atlasskolstvi.cz“. Bohužel velká část zde uvedených kontaktů již není funkční. Ve většině případů na můj email nikdo nereagoval. Velkou část potenciálních respondentů jsem ztratil díky spamovým filtrům. Ve výsledku mám 401 respondentů.

V dotazníku mě nejvíce zajímalo, zda se vyučuje na základních školách programování. Pokud respondent odpověděl „ano“, následovala otázka na ročník, ve kterém programování vyučují. Dále mě zajímala časová dotace a používané vývojové prostředí. Poslední otázkou na tuto skupinu dotázaných bylo, jestli by uvažovali o změně vývojového prostředí a za jakých podmínek.

Pokud respondent odpověděl „ne“, ptal jsem se na důvod této skutečnosti. Tato skupina respondentů mě zajímala nejvíce, a proto má další otázka směřovala k tomu, jestli by byli ochotni zařadit programování do výuky informatiky a za jakých podmínek.

2.2 Výsledky dotazníkového šetření

Vyučujete v rámci informatiky na základní škole také základy programování?



Graf 2.1: Vyučujete v rámci informatiky na základní škole také základy programování

V kolikátém ročníku programování vyučujete?



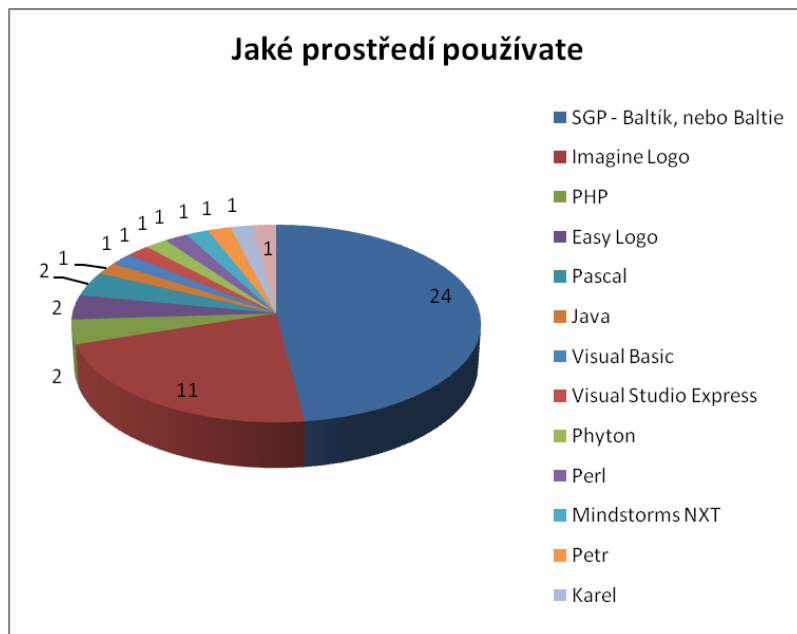
Graf 2.2: V kolikátém ročníku programování vyučujete

Respondenti mohli vybrat více než jeden ročník, takže počet v součtu překračuje hranici 57 odpovědí. Z výzkumu vyplynulo, že nejvíce se na základních školách žáci seznamují s programováním právě v deváté třídě. Nechal bych na zvážení každého učitele, kdy zařadí programování do výuky informatiky. Nicméně Pecinovský ve svém článku [7] vidí jako ideální věk žáků pátou až šestou třídu.

S jakou hodinovou dotací?

Na tuto otázku odpovídali respondenti velice různorodě. Objevují se různé časové dotace od pěti do nesmyslných sto hodin. Nejčastější odpovědi byly „10 hodin“, „1 hodinu týdně“ a „2 hodiny týdně asi 3 měsíce“.

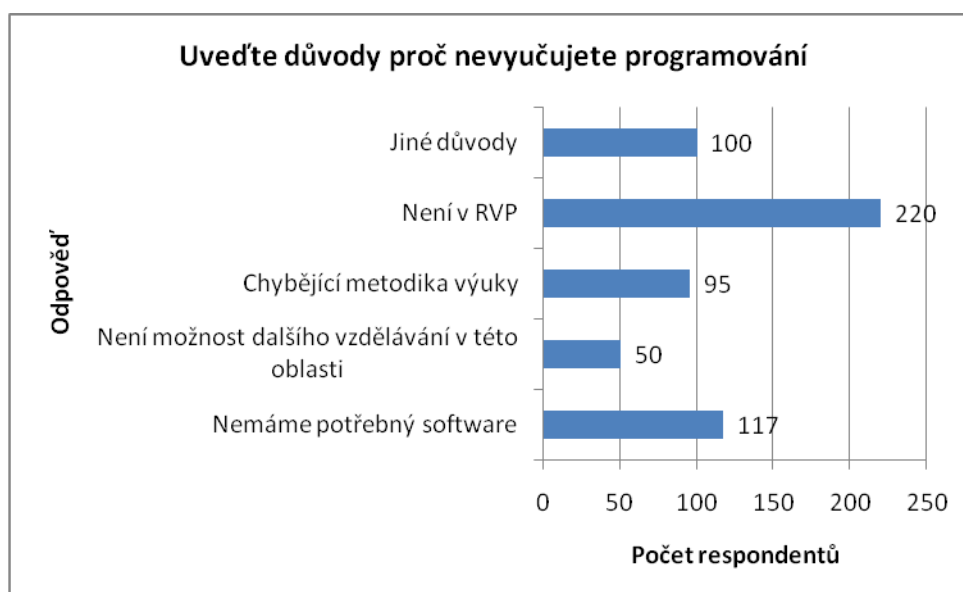
Jaké prostředí používáte?



Graf 2.3: Jaké prostředí používáte

Njevíce je v odpovědích zastoupen program Blatík nebo Baltie a to počtem 24. Imagine Logo používá pouze 11 z 57 respondentů. Největším překvapením pro mě je zastoupení ostatních jazyků nebo programů. Byly uváděny například jazyky PHP, Java, Perl, Python, Visual Basic, Pascal, a prostředí Visual Express Studio 2008, Petr, Karel, Easy Logo, Mindstorms NXT. Do grafu jsem nezahrnul odpovědi jako například Textový editor, Excel, HTML, „žádný“ atp.

V případě, že respondenti odpověděli „NE“, uváděli důvody



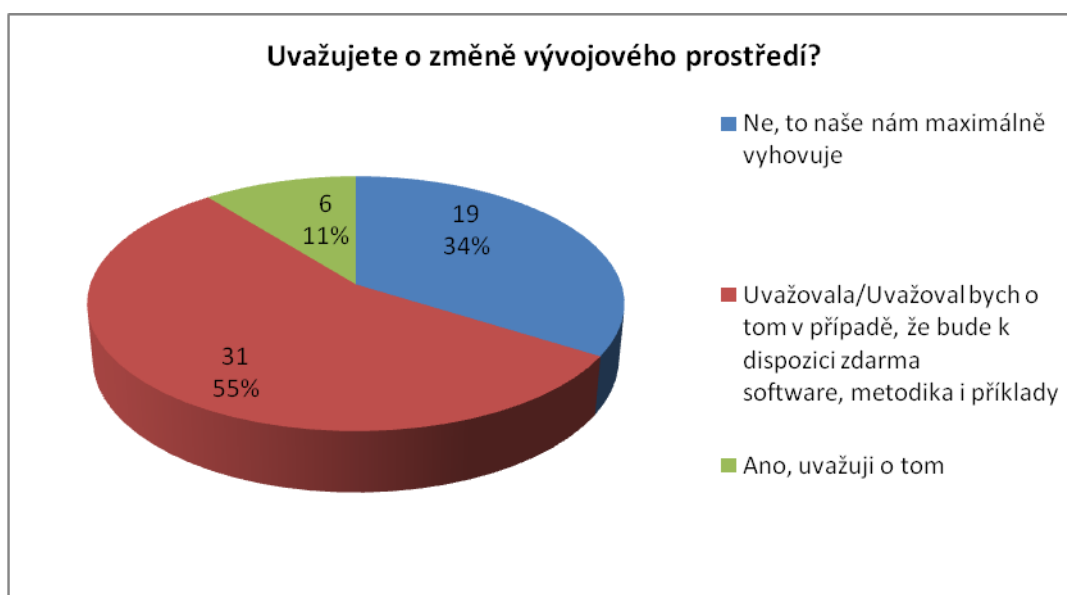
Graf 2.4: Důvody proč neučit na ZŠ informatiku, které uvedli učitelé

Respondenti mohli vybrat více než jeden důvod, takže počet v součtu překračuje hranici 344 odpovědí. Zde se potvrdil můj předpoklad, nejčastěji se učitelé odvolávají na Rámcový vzdělávací program. Velmi mě překvapily některé ostatní důvody, jako například podceňování žáků nebo důvod že žáci potřebují získat spousty jiných kompetencí.

Respondenti uváděli například:

- Nedostatečná časová dotace výuky informačních a komunikačních technologií (IKT), toto byl nejčastější důvod.
- Žáci na ZŠ potřebují získat v oblasti IKT spoustu jiných kompetencí...
- Žáci naší školy by toto učivo nezvládli.
- Priority vidíme v jiných oblastech vzdělávání.
- Obtížnost výuky, složité.
- V běžné škole pro většinu dětí nemá význam.
- Učíme jiná témata - 3D modeling a podobně.
- Není vyučující (aprobace).
- Je nad možnosti žáků.
- Programování necháváme na SŠ.
- Vyučovat plošně na základní škole v 5-7 třídách programování je scestné.

Uvažujete o změně vývojového prostředí?



Graf 2.5: Uvažujete o změně vývojového prostředí

Z těchto odpovědí vyplývá, že má práce by mohla nalézt uplatnění při výuce i na školách, kde již nějaké prostředí pro výuku programování používají.

Uvažujete o zařazení programování do výuky informatiky?



Graf 2.6: Uvažujete o zařazení programování do výuky informatiky

Toto zjištění je pro mě stěžejní. Osm procent respondentů uvedlo, že o výuce programování uvažují. Téměř polovina z učitelů, kteří zatím programování neučí, by o tom uvažovalo, pokud by měli k dispozici vše, co pro výuku programování na základní škole potřebují.

2.3 Shrnutí výsledků průzkumu

Z mého průzkumu vyplývá, že téměř polovina učitelů, kteří zatím programování neučí, čeká na vhodnou příležitost, aby jej do výuky zařadili. Právě to dává mé práci určitý potenciál, že by mohla být ve výuce informatiky na základních školách přínosná. Dokonce někteří učitelé mě kontaktovali emailem mimo můj dotazník a vyslovili zájem o mou práci. Právě tyto učitele bych nerad zklamal a pokusím se vytvořit materiál, který jim bude přínosný.

3. Porovnání programů pro výuku programování na ZŠ

Na základě provedeného průzkumu provedu srovnání dvou nejvíce používaných prostředí pro výuku programování s prostředím Scratch. Nejpoužívanější prostředí jsou Baltík a Imagine Logo. Na programy budu nahlížet z pohledu několika kritérií. Nejvíce mě zajímají náklady a způsob pořízení programů, možnosti programu v oblasti výuky, komfort a uživatelská přívětivost prostředí, snadnost zápisu a editace algoritmů, práce s daty, práce s grafikou, práce s objekty a intuitivnost prostředí.

3.1 Baltík

Program Baltík ve verzi 3.7 jsem stáhnul ze stránek firmy SGP. Jedná se o zkušební verzi, která je omezena. Takto nainstalovaný program mi neumožnil ukládat skripty ani scény.

Pořízení programu

Program lze zakoupit přímo od výrobce. Jsou k dispozici různé varianty od ročních licencí až po trvalé. Například trvalá licence na počítače v celé škole vyjde na 25 000 Kč. Roční licence na všechny počítače ve škole stojí 3500 Kč. Učebnice, jejímž autorem je pan Pecinovský, potom vyjde na 199 Kč. Program může být instalován na serveru ve sdílené složce a přístupný všem uživatelům pro spuštění ve školní síti.

Možnosti programu

Program Baltík používá syntaxi jazyka C. Možnosti programu pro výuku jsou široké. Program obsahuje několik výukových režimů – skládat scénu, čarovat scénu, programovat začátečník a programovat pokročilí. Každý režim je určen pro určitou věkovou kategorii dětí. Mě zajímal režim „programovat pokročilí“, kde jsou k dispozici všechny klasické příkazy. V programu skládáme příkazy, které pak Baltík provádí ve scéně. Scénu můžeme upravit a libovolně poskládat z předmětů. Tato možnost byla pro mě nepřístupná, protože jsem ve zkušební verzi neměl možnost uložit scénu do souboru a později ji vyvolat. Je zde k dispozici knihovna předmětů, které můžeme libovolně upravit. Program Baltík obsahuje všechny základní příkazy jazyka C (if, for, while, do-while, atd.), snad jen kromě příkazu switch. Je možné používat proměnné, konstanty a pole různých datových typů. Jsou zde k dispozici také procedury a funkce (pomocníci). Troufám si tvrdit, že pomocí Baltíka by bylo možné učit všechna témata jazyka C například podle učebnice [20], včetně ukazatelů a témat s nimi spojenými.

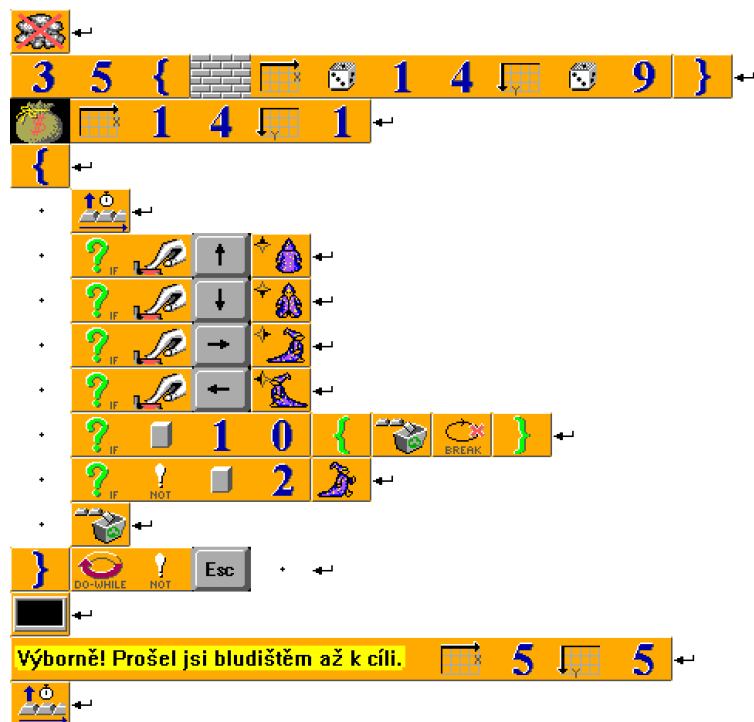
Navzdory tomu, že v programu nemusí žáci psát příkazy, jen je přetahují do plochy skriptů, jsem velmi zápasil s rozpoznáním jednotlivých příkazů. Vždy jsem musel kliknout na nápovědu u příkazu a až pak jsem pochopil, co daný symbol znamená. Tohle mi na programu vadilo a dost mě to zdržovalo při vytvoření mého vzorového programu bludiště. Pokud si na to učitel, potažmo žáci zvyknou, potom už jim to jistě nečiní potíže. Nápověda je dobře zpracovaná a vždy jsem z ní pochopil, jak příkaz funguje.

Při editaci kódu lze označit příkaz, řádek příkazů, nebo určitý blok příkazů a dále je kopírovat, vkládat i mazat. Dostal jsem se do situací, kdy nebylo možné označit určitou část příkazů a musel jsem je přetahovat po jednom. Z těchto důvodů mi nepřijde Baltík uživatelsky komfortní, ale věřím, že je to jen o zažitých návycích.

Práce s daty se velmi podobá reálným jazykům a je možné používat proměnné, konstanty a pole různých datových typů. Grafiku scény si můžeme libovolně navrhnout, nebo vykouzlit pomocí předmětů. Kreslit s Baltíkem ve scéně tak, jak je to možné v Imagine nebo Scratch, tady nelze.

Baltík není zaměřen na objektové programování. Objektové programování se žáci mají možnost učit v programu Baltie, který se velmi podobá programovacímu jazyku C#.

Ukázka zdrojového kódu:



Obrázek 3.1: Ukázka zdrojového kódu programu Baltík

3.2 Imagine Logo

Program Imagine Logo jsme používali při výuce předmětu Didaktika informatiky na PF JU s Doc. Vaníčkem. Díky tomu mám k dispozici plnou verzi programu. Jinak demoverzi programu je možné stáhnout ze stránek [22]. Demoverze, stejně jako v případě Baltíka, neumožňuje ukládat vytvořené programy.

Pořízení programu

Imagine lze koupit u nakladatelství Computer Press, které je distributorem programu v ČR, ve třech licenčních variantách [21]. Domácí licenci lze zakoupit za 690,- Kč. Omezená školní licence, která stojí 4 760,- Kč, je určena k použití na maximálně 20 počítačích. Neomezená školní licence stojí 9 520,- Kč. Velkou výhodou programu je, že pokud škola vlastní školní licenci Imagine Logo, její žáci a učitelé mohou program v plné verzi bezplatně používat také doma.

Možnosti programu

Imagine je kompletně objektový jazyk, který je řízen událostmi. Podporuje paralelní programování a má též propracovanou ideu obrázkových tvarů želv. Používá prvky, které jsou typické pro programy pod Windows, např. překrývající se grafické plochy (jako listy papíru), tlačítka i s obrázky, posuvné lišty, textová pole, lišty tlačítek atd. Nechybějí ani multimédia, Internet a též vzájemná spolupráce Imagine-programů v síti. [22]

Malou nevýhodou programu Imagine může být, že všechny příkazy se musejí napsat slovem. Příkazy zde nejsou reprezentovány symboly, jako je tomu u programů Baltík a Scratch. Také zde nelze programovat intuitivně. Je potřeba alespoň minimální znalost příkazů a další dohledávat. Naštěstí je k dispozici on-line zdroj [19], kde není problém se základy naučit.

Ukázka zdrojového kódu:

```
příkaz pohyb
když klávesa?
[
  vPřípadě klávesa
  [
    Esc [ukonči]
    nahoru [směr! 0 do 2 dosad "a barvabodu když :a="černá [vz 4]]
    vpravo [směr! 90 do 2 dosad "a barvabodu když :a="černá [vz 4]]
    dolů [směr! 180 do 2 dosad "a barvabodu když :a="černá [vz 4]]
    ulevo [směr! 270 do 2 dosad "a barvabodu když :a="černá [vz 4]]
  ]
  dosad "a barvabodu když :a="červená [ž2'ukaž dosad "klidnej 1 ukonči]
pohyb
konec
```

Obrázek 3.2: Ukázka zdrojového kódu programu Imagine

3.3 Scratch

Program Scratch, který lze stáhnout z [23] je ve verzi 1.4 a je nutné ho nainstalovat do počítače. Novinkou je cloudové řešení programu Scratch 2.0, které je dostupné přímo přes webový prohlížeč, nyní již v plné verzi [24].

Pořízení programu

V tomto ohledu má Scratch ve srovnání všech tří programů obrovskou výhodu. Nejenom, že je zdarma, ale obrovský potenciál skýtá cloudové řešení, které je všem žákům, učitelům a dalším vždy k dispozici a přístupné přes internetový prohlížeč. Stačí jen spustit příslušnou webovou stránku, přihlásit se (není podmínkou) a začít programovat. Bohužel toto řešení má jednu nevýhodu a to v případě výpadku internetu, program nebude dostupný.

Možnosti programu

Program Scratch 2.0 obsahuje oproti verzi 1.4 také celou řadu novinek a vylepšení. Jednou z nejvýznamnějších novinek je možnost naprogramovat proceduru a funkci, kterou doposud nebylo možné používat. Další vylepšení se týká klonování postav. Velmi oceňuji přepracovaný Paint editor, který nově může pracovat ve dvou režimech – vektorová nebo bitmapová grafika.

Scratch je jednoduché výukové programovací prostředí umožňující programovat za pomoci bloků příkazů, které se do sebe skládají podobně jako puzzle. Díky této vlastnosti je Scratch ve srovnání s ostatními programy nejintuitivnější a jednodušší na ovládání. Tímto se také odbouraly syntaktické chyby. V programu Imagine můžeme napsat libovolné příkazy a v jakémkoli pořadí. To samé platí u Baltíka, také lze přesunout příkazy kamkoliv a na první pohled nepoznáme, že takto zapsané příkazy jsou špatně. V programu Scratch ihned vidíme, že pokud nám příkazy do sebe nezapadnou, nelze je takto použít.

Velmi pěkná je také názornost samotného scénáře ve Scratch. Všechny příkazy jsou na samostatném řádku a velmi přehledně do sebe zapadají, případně se příkazy do sebe vnořují. Tuto vlastnost jsem velmi postrádal u Baltíka, kde jsem si musel dát velmi záležet na ruční úpravě kódu, abych se v něm sám vyznal.

Program Scratch umožní žákům seznámení se všemi základními prvky programování. Zápis a úprava algoritmů je zde jednodušší než v programech Baltík a Imagine. Ovšem Scratch na rozdíl od Baltíka neřeší datové typy u proměnných a konstant. Otázkou zůstává, jestli je to při výuce programování na základních školách nutné téma. Například programátor v jazyce PHP se datovými typy proměnných také nemusí zabývat. Scratch také neumožňuje plnohodnotnou práci s objekty, tak jak to umožňuje program Imagine. Například nelze ve Scratch pracovat se třídami.

Pracovat s daty v programu Scratch můžeme na úrovni proměnných a polí. Co se týče práce s grafikou, je zde patrná podobnost s programem Imagine. Oba programy v tomto ohledu předčí Baltíka.

Ukázka zdrojového kódu:



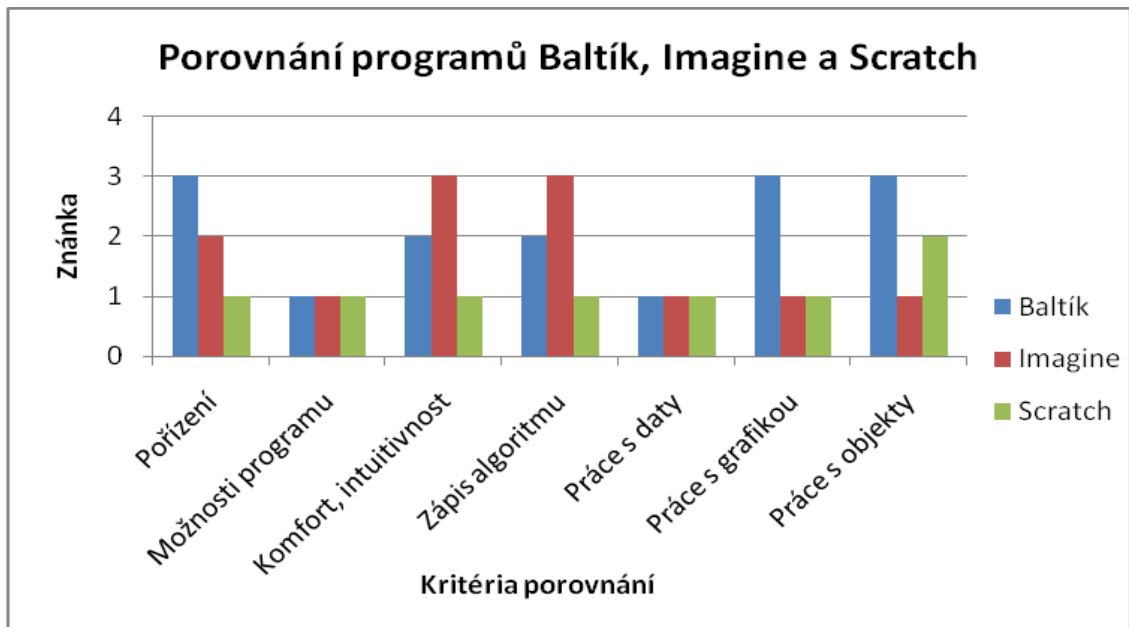
Obrázek 3.3: Ukázka zdrojového kódu programu Scratch

3.4 Porovnání programů

Pro porovnání programů jsem použil tato kritéria:

1. Pořízení (cena, instalace)
2. Možnosti programu v oblasti výuky
3. Komfort, intuitivnost a uživatelská přívětivost prostředí
4. Snadnost zápisu a úpravy algoritmu
5. Práce s daty
6. Práce s grafikou
7. Práce s objekty

Mým cílem není kritizovat programy, které jsou uznávané a používají se při výuce programování na základních školách. Všechny programy jsou kvalitní a každý má své výhody. Mé hodnocení je subjektivní a vychází ze zkušeností, které s programy mám. Porovnání jsem provedl tak, že každý program dostal v daném kritériu známku „1“ až „3“. Zámka „1“ je nejlepší a známka „3“ nejhorší.



Graf 3.4: Porovnání programů Baltík, Imagine a Scratch

4. Materiály dostupné k výuce programování na ZŠ

Abych mohl vytvořit vlastní postup výuky, prostudoval jsem materiály, které se používají při výuce programování na základních školách. Vzhledem k výsledkům průzkumu jsem zde zařadil jeden materiál zaměřený na výuku programování v SGP Baltík, učebnici zaměřenou na programování v Imagine a materiály dostupné pro výuku Scratch. Porovnám zde, jaký postup výuky autoři volí v jednotlivých prostředích.

4.1 Tvořivá informatika s Baltíkem

Materiálů pro výuku Baltíka se mi nepodařilo najít mnoho. Přímou na stránkách SGP jsou k dispozici na stažení různorodé osnovy, které zpracovali učitelé. Existuje učebnice, kterou prodává přímo firma SGP, ale tuto učebnici jsem bohužel neměl k dispozici a není dostupná ani v knihovně. Ovšem podařilo se mi najít zdroj informací dostupný přímo na internetu. Jsou to metodické materiály, které byly vytvořeny v rámci projektu a jsou všem k dispozici na stažení zdarma. Autorkou materiálů je Eva Hlavatá. Metodické materiály jsou určeny pro výuku programování v programovacím jazyce Baltík 3 a Baltie 4. Mohou sloužit učitelům přímo jako příprava na hodinu. Součástí materiálů jsou i zadání s jejich řešením v elektronické podobě. Zde uvedu pouze materiály vytvořené k programu Baltík 3, protože obsahem výuky se více blíží programu Scratch. Baltie 4 se výrazněji odlišuje od programu Scratch tím, že zobrazuje scénu ve 3D a také se více zaměřuje na objektově orientované programování. [16]

Obsah výuky:

Tvořivá informatika s Baltíkem I:

- seznámení s programem
- posloupnost příkazů
- opakování příkazů.
- použití procedury (pomocníka)
- vytvoření vlastních předmětů v Paint editoru a vložení zvuků do programu.

Tvořivá informatika s Baltíkem II:

- rozhodování
- cyklus s podmínkou na začátku
- proměnné

- konstanty

Tvořivá informatika s Baltíkem III:

- lokální proměnné a funkce s parametry
- cyklus s pevným počtem opakování
- využití matematických funkcí
- složené podmínky a logické operátory

4.2 Imagine Logo Učebnice programování pro děti

Pro podporu výuku Imagine Logo jsem našel tři významnější zdroje. Mnoho informací lze nalézt v učebnici [6] a také na webových stránkách [19]. Zde uvedu učebnici „Imagine Logo Učebnice programování pro děti”. Autory této učebnice jsou Andrej Blaho a Ivan Kalaš. Učebnice je určena především k výuce programování na základních školách v prostředí Imagine Logo. Do své analýzy jsem ji přesto zahrnul a to ze dvou důvodů. Jedním z důvodů je původ programu Scratch, který vychází právě z prostředí Imagine Logo. Druhým důvodem je použitelnost této knihy k výuce programování v programu Scratch.

Obsah kapitol:

1. Pero, barva a tloušťka – kapitola je zaměřena na posloupnost příkazů.
2. Opakuj a pojmenuj – příkaz cyklu, nový příkaz (procedura), použití nového příkazu ve svém vlastním programu.
3. Stavebnice příkazů – procedury, skládání procedur do větších celků
4. Různé události v životě želvy – definice událostí pro objekt (želvu).
5. Více želv více možností – žáci se učí vytvořit další objekt (želvu), která má své události a své příkazy.
6. Želvy a jejich tvar – vlastnosti a události objektu (želvy).
7. Animované tvary a procesy – žáci se naučí pomocí příkazu „každých“ vytvořit procesy, vytvořit a použít animovaný tvar, používat souřadnice stránky.
8. Příkazy s proměnnými – definování a použití funkce.
9. Pohyby a závody – podmíněný příkaz „když“ a procesy, proměnné.
10. Pokusy a další hry – poslední kapitola obsahuje postup vytvoření her, na kterých si žáci procvičují naučené příkazy.

Analýza

Učebnice obsahuje velké množství příkladů, na kterých žáci procvičují probíraný příkaz. Žáci od začátku výuky tvoří programy. Autoři nejprve vyloží určitý prvek jazyka a ten následně procvičují na daných příkladech. Učebnice tím splňuje téměř všechny zásady výuky dle [4]. V programu Scratch však většina učitelů volí jiný postup výuky, než jak ho ukazuje tato učebnice. Pro porovnání této učebnice s postupy výuky ve Scratch si zvolím dva zástupce, a to [12] a [15]. Materiály se shodují v prvních kapitolách, kde se učí posloupnost příkazů. Také na zařazení cyklu do výuky se učebnice shoduje s materiály zpracovanými přímo pro Scratch. Zcela zásadní rozdíl je ve druhé kapitole, kde Blaho a Kalaš zařazují do výuky procedury. Tento přístup se ve výuce Scratch [12] a [15] nepoužívá. Do nedávna tento postup ani nebyl možný, protože Scratch procedury a funkce neobsahoval. Ačkoliv nejsou ostatní kapitoly učebnice identické s [12] a [15], obsahem vyučovaných prvků jsou si velice podobné. Vzhledem k podobnosti obou programů by mohla být učebnice určitým vodítkem pro výuku programování ve Scratch a to za předpokladu, že by se upravilo pořadí jednotlivých kapitol.

4.3 Bakalářská práce Možnosti využití jazyka Scratch na ZŠ

Bakalářská práce, jejíž autorkou je Veronika Nováková, není veřejně dostupná. Autorka mi ji poslala na základě mé prosby. Autorka se ve své bakalářské práci věnuje otázce zařazení a postavení výuky programování na základních školách a analyzuje možnosti využití programovacích prostředí jako Logo, ComeniusLogo, Imagine, Karel, Baltík a Baltazar, které se používali a stále ještě používají na některých českých základních školách. Autorka seznamuje s principy programování programovacího prostředí Scratch 1.3 a 1.4 pro tvorbu multimediálních úloh. Součástí bakalářské práce je i rozbor a hodnocení ukázek dostupných na Internetu a návrhy souboru úloh, projektů a námětů pro žáky.

Obsah kapitol

1. Pohyb – posloupnost příkazů.
2. Vzhled – cyklus s počtem opakování, vlastnosti objektu.
3. Zvuk – cyklus s daným počtem opakování, vlastnosti objektu, posloupnost příkazů.

4. Pero – kreslení za pomoci postavy – posloupnost příkazů, vlastnosti objektu, souřadnice scény, projekt Smyčka – cyklus s daným počtem opakování, vnořený cyklus.
5. Vnímání a Operátory – nekonečný cyklus, podmíněný příkaz, operátor porovnání rovnosti, posílání zpráv mezi objekty.
6. Operátory a proměnné – práce s proměnnými, cyklus s podmíněným počtem opakování, logické a aritmetické operátory, podmíněný příkaz, posílání zpráv mezi objekty.

Analýza

Všechny kapitoly jsou postavené tak, že obsahují nejméně jednu ukázkovou úlohu a nejméně jednu samostatnou úlohu pro žáky na procvičení daného tématu. Od první úlohy jsou k výuce používány dvě postavy, ale žádná úloha neřeší, jak přidat další objekty (postavy ani scény) a jak s nimi pracovat. V průběhu všech úloh jsou využívány události postavy. Autorka ve svých úlohách seznamuje žáky s důležitými pojmy, jako jsou proměnné, podmíněný příkaz, cyklus a operátory. Žádný z uváděných příkladů není zaměřen na testování a ladění. V úlohách také chybí prostor pro tvořivost žáků, která je podle Mikolajové [6] jednou z dílčích cílů výuky. Příklady také postrádají složitější problémy, které by žáci sami analyzovali, rozdělili je na dílčí problémy a ty řešili [4].

4.4 Materiály pro Scratch dostupné na webu

Při přípravě své diplomové práce jsem prostudoval mnoho materiálů, které jsou dostupné na portálu scratched.media.mit.edu. Na webu jsem velice snadno nelezl zdroje pro učitele, které jsou určené pro základní školu a jsou anglicky. Pro srovnání jsem vybral několik zdrojů, na které učitelé při hledání dostupných materiálů mohou narazit. Ve svém srovnání uvedu jen krátký popis k jednotlivým lekcím a zanalyzuji postup výuky.

4.4.1 Learn Scratch

Autor: Dan Hawk [10]

Autor v osmi lekcích popisuje prostředí Scratch a jeho příkazy. Velice zajímavý je úvod, v němž jsou srovnávány příkazy jazyka Java s příkazy (bloky) Scratch a tím se ukazuje jednoduchost programu Scratch.

Obsah lekcí:

1. lekce – vlastnosti a události objektu (postavy).
2. lekce – vlastnosti a události objektu (postavy), operátory, cyklus s podmíněným počtem opakování.
3. lekce – vlastnosti objektu (pozadí), cyklus s daným počtem opakování.
4. lekce – proměnné.
5. lekce – procvičení naučených příkazů, cykly s podmíněným počtem opakování, souřadnice scény.
6. lekce – cyklus s podmíněným počtem opakování, operátory, souřadnice scény.
7. lekce – posílání zpráv mezi objekty.
8. lekce – vlastnosti objektu – přidání zvuků postavě.

Analýza

Obsah výuky se od první lekce neshoduje s pojetím výuky v ostatních materiálech, jako například [15]. Autor nezačíná výuku posloupností příkazů, ale rovnou seznamuje s vlastnostmi a událostmi objektu. Celkový obsah výuky je přizpůsoben vytvoření jedné konkrétní hry a na to jsou navázány ukázky konkrétních příkazů. Hawk také nezařazuje do výuky žádné jiné příklady, aby si žáci osvojili použití důležitých programátorských konceptů [4].

Tento zdroj může posloužit učitelům, kteří se chtějí seznámit se základy práce v tomto prostředí, ale není vhodnou metodikou výuky programování.

4.4.2 Introduction to Programming

Autorka: Amanda Wilson [11]

Autorka zpracovala plán a pracovní listy na výuku v prostředí Scratch, který obsahuje osm lekcí.

Obsah lekcí:

1. lekce – seznámení s prostředím a prací ve Scratch.
2. lekce – posloupnost příkazů.
3. lekce – posloupnost příkazů.
4. lekce – cyklus s daným počtem opakování.
5. lekce – podmíněný příkaz.

6. lekce – posílání zpráv mezi objekty, cyklus s podmíněným počtem opakování a operátory.

7. lekce – orientace ve zdrojovém kódu.

8. lekce – test.

Analýza

Autorka se ve svých lekcích věnuje základním prvkům programování, ale opomíjí problematiku proměnných. Neuvádí také příklady, na kterých by si žáci osvojili použití důležitých programátorských konceptů. Naopak velmi dobře zařadila autorka do výuky orientaci v neznámém zdrojovém kódu. Žáci naprogramují část úlohy, tu si mezi sebou vymění a úlohu dokončí. Materiály obsahují plán lekcí a žákovské pracovní listy, které by mohly posloužit jako doplněk výuky Scratch. Seminář, ke kterému autorka připravila tyto materiály, slouží jako základní seznámení žáků s prostředím Scratch.

4.4.3 Scratch Lessons: Shall We Learn Scratch Programming for Tweens

Autorka: Jessica Chiang [12]

Autorka zpracovala obsáhlé materiály na výuku v prostředí Scratch, které jsou rozděleny do patnácti lekcí.

Obsah lekcí:

1. lekce – seznámení s prostředím Scratch, obrázkový editor, malování postavy

2. lekce – cykly (nekonečný a s daným počtem opakování), vlastnosti objektu.

3. lekce – nekonečný cyklus a vlastnosti objektu (postavy).

4. lekce – proměnné a operátory, přidání další postavy do scény.

5. lekce – události objektu (postavy) a souřadnice ve scéně.

6. lekce – vlastnosti objektu – přidání zvuku, import zvuků (hudby).

7. lekce – cykly, události objektu a operátory – hra „Pong“

8. a 9. lekce – posílání zpráv mezi objekty – tvorba animovaného příběhu.

10. - 15. lekce – žáci procvičují proměnné, podmíněný příkaz, operátory a cykly při tvorbě složitější hry – plánují scénář hry, malují postavy a scény, programují scénáře.

Analýza

Jednotlivé lekce obsahují seznámení se všemi důležitými příkazy (bloky) programovacího prostředí Scratch a sled výuky jednotlivých témat je téměř shodný s [15]. Autorka na závěr výuky zařadila složitější projekt – hru Mario, tím naplňuje

jednu ze zásad výuky programování [4] a předkládá žákům složitější projekt, který musí na základě svých znalostí řešit.

4.4.4 Resources for teachers, parents and computer club facilitators

Autor: autoři serveru redware.com [13]

Na serveru je dostupných devět lekcí, ve kterých autoři ukazují učitelům jak pracovat s vývojovým prostředím Scratch. Každá lekce obsahuje krátký slovní doprovod a většina lekcí obsahuje video, které ukazuje práci se Scratch.

Obsah lekcí:

1. lekce – představuje obsah kurzu.
2. lekce – práce s obrázkovým editorem a malování postavy.
3. lekce – posloupnost příkazů, události, proměnné, operátory, cykly s daným počtem opakování – kreslení pomocí postavy ve scéně.
4. lekce – souřadnice scény, události objektu (postavy), nekonečný cyklus.
5. lekce – zvuky a grafika. Video ukázka je momentálně nefunkční.
6. lekce – proměnné.
7. lekce – posílání zpráv mezi objekty.
8. lekce – tvorba her, procvičení příkazů z předchozích lekcí.
9. lekce – tvorba multimediálních prezentací ve Scratch.

Analýza

Video tutoriály postupně seznamují s bloky příkazů. Nejobsáhlejší je třetí lekce, ve které autor během osmi minut ukáže práci s událostmi, proměnnými, cykly a operátory. Tento postup může seznámit učitele s problematikou, ale pro výuku programování je tato ukázka nevhodná. Ukázka obsahuje zbytečně složité příklady a všechny důležité prvky programování jsou zde používány najednou. Materiály také neobsahují žádné příklady na procvičení, jak doporučuje Pecinovský [4]. Materiál učiteli může pomoci při seznámení se Scratch. Pokud by učitel chtěl postupovat při výuce podle uváděných postupů v jednotlivých lekcích, musí na procvičení použít vlastní příklady a především třetí lekci rozdělit na několik lekcí.

4.4.5 LearnScratch

Autor: učitelé, studenti a zaměstnanci univerzity La Salle [14]

Na portálu learnscratch.org je dostupné obrovské množství videí, která jsou rozdělena do tří kurzů:

Scratch 1

Kurz obsahuje 8 lekcí – seznamuje se základní prací v programu Scratch.

1. lekce – posloupnost příkazů.
2. lekce – opakování.
3. lekce – vlastnosti objektu – změna efektu postavy.
4. lekce – přidání dalšího objektu (postavy).
5. lekce – výstup – příkaz „říkej“.
6. lekce – vlastnosti objektu – změna efektu obrázku.
7. lekce – vlastnosti objektu – přidání zvuku postavě
8. lekce – vlastnosti objektu, posloupnost příkazů, opakování – animace postavy

Každá lekce obsahuje video ukázkou k danému tématu a několik návrhů na cvičení (například „Experimentuj se změnou kostýmů postavy a vytvoř animaci.“).

Scratch 2

Kurz obsahuje celkem 24 lekcí, které jsou rozděleny do pěti částí. Tento videokurz seznamuje s jednotlivými bloky příkazů postupně ve všech kategoriích (Pohyb, Vzhled, Zvuk, Pero, Ovládání, Vnímání, Operátory a Proměnné). V každé video ukázce je předvedena práce s bloky příkazů a ukázky použití. Každá lekce obsahuje návrhy na samostatné procvičení příkazů.

Scratch 3

Kurz obsahuje 7 částí a celkem 32 lekcí. Lekce jsou založeny na souboru projektů převzatých od uživatelů Scratch. V tomto kurzu mohou učitelé hledat inspiraci pro žákovské projekty.

Analýza

Materiál Scratch 1 obsahuje postup výuky, ale jen zlomek důležitých programátorských prvků – posloupnost příkazů, cyklus a vlastnosti objektu. Ostatní zde nejsou obsaženy.

Materiál Scratch 2 není vhodným postupem výuky. Jsou zde sice probírány všechny příkazy Scratch, ale jsou procházeny postupně po kategoriích. Obsah lekcí se

nezaměřuje na plánování, postupný vývoj, rozdělení problému na menší části, testování ani ladění.

Materiál Scratch 3 je jen souborem projektů a neobsahuje žádný postup výuky.

4.4.6 Scratch curriculum guide

Autoři: Karen Brennan, Jeff Hawson a Michelle Chung [15]

Jedná se o metodiku výuky programování v prostředí Scratch. Materiál je rozdělen do pěti hlavních témat a obsahuje 20 lekcí.:

Úvod

1. lekce – žáci se seznámí s prostředím programu Scratch.
2. lekce – samostatné zkoumání prostředí Scratch – žáci vytvoří životopisný projekt na základě ukázkových projektů.

Umění

3. lekce – posloupnost příkazů. (bez PC – dvojice žáků – jeden z žáků bude na základě shlédnuté video ukázky vydávat slovní příkazy a druhý je bude vykonávat)
4. lekce – posloupnost příkazů, cyklus s daným počtem opakování, vlastnosti objektu, přidání dalšího objektu (postavy), postupný vývoj a testování.
5. lekce – procvičení sekvencí, cyklů, osvojení postupného vývoje, testování a ladění, abstrakce a rozdělení problému na menší části.

Příběhy

6. lekce – paralelizace, události.
7. lekce – paralelizace a událostí, znovupoužití již známých příkazů, orientace v cizím zdrojovém kódu, týmová spolupráce.
8. lekce – posílání zpráv mezi objekty, paralelizace, události, znovupoužití již známých příkazů, testování a ladění.

Hry

9. lekce – testování a ladění.
10. lekce – podmíněný příkaz a operátory
11. lekce – proměnné – žáci ve trojicích samostatně zkoumají ukázkové projekty, potom to naučí také ostatní
12. lekce – osvojení podmínek, operátorů a proměnných, znovupoužití již známých postupů.

Závěrečný projekt

13. lekce – žáci vytvářejí návrh vlastního projektu a osnovu, plánování.

14. – 19. lekce – vývoj vlastního projektu v týmu – sekvence příkazů, smyčky, paralelizace, události, podmínky, operátory, proměnné, testování a ladění (zpětná vazba – v 16. lekci je zařazeno kritické hodnocení projektu ostatními)

20. lekce – prezentování projektu před ostatními

Analýza

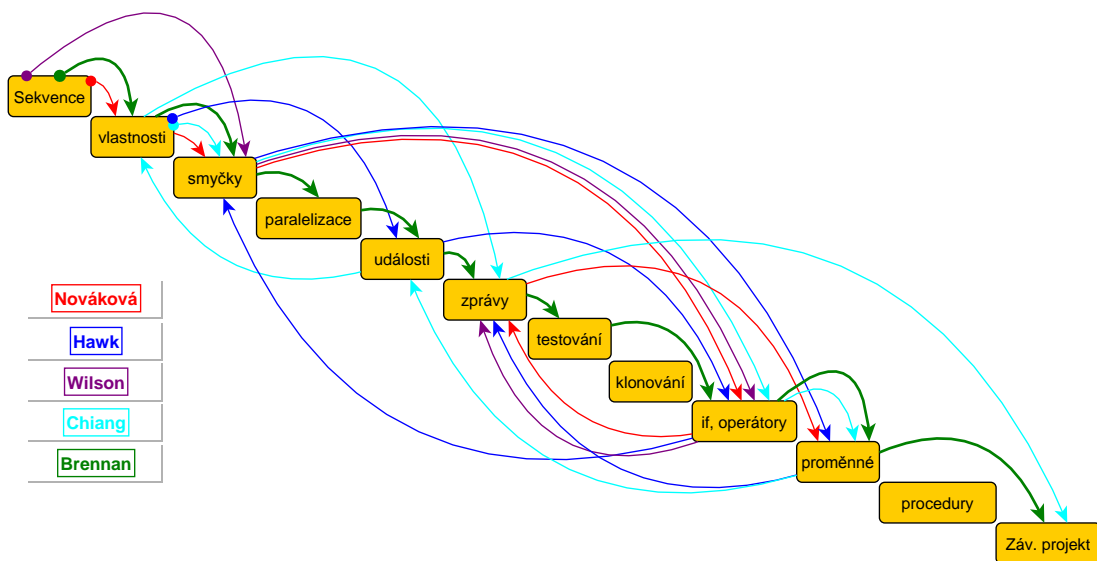
Tento materiál se zaměřuje na rozvoj programátorského myšlení a počítačové tvořivosti žáků pomocí zkoumání programu Scratch. Způsobem zpracování se od ostatních zdrojů naprosto odlišuje. V úvodu je v tabulkách uveden přehled hodin s prováděnými činnostmi žáků, programátorské pojmy a postupy, které se během lekcí žáci učí. Jednotlivé přípravy na hodiny obsahují popis hodiny, cíle hodiny, souhrn aktivit, zdroje a podrobný popis hodiny. Velký důraz klade autorka na plánování a přemýšlení nad daným problémem a následné zapsání svých myšlenek. Výuka v jednotlivých lekcích splňuje zásady správné výuky programování dle [4], například žáci tvoří programy od začátku výuky, příkazy jsou probírány postupně „po malých soustech“, příklady jsou zajímavé a žáci se učí své programy ladit. Dále tato výuka u žáků rozvíjí kreativní myšlení, schopnosti analyzovat a plánovat, rozdělit si problém na menší a snáze řešitelné podproblémy. Všechny tyto důležité návyky vidí [8] jako nezbytné cíle výuky programování. V jednotlivých lekcích jsou probrány všechny důležité prvky programování – sekvence příkazů, smyčky, paralelizace, vlastnosti objektů, události, posílání zpráv mezi objekty, podmínky, operátory, proměnné, orientace v neznámém zdrojovém kódu, testování a ladění.

4.5 Shrnutí

Zdroje dostupné z portálu scratched.media.mit.edu, které zde uvádím, ukazují práci v prostředí Scratch. Většina z nich neobsahuje témata zaměřená na testování a ladění, a také dostatek příkladů na procvičení probíraných témat. Autorka zdroje [11] dokonce opomíjí téma proměnných. V těchto případech se nejedná o metodiku výuky, která by postupně prohlubovala znalosti programování, plánování, myšlení a tvořivost u žáků. Většina materiálů si neklade za cíl rozvíjet u žáků klíčové kompetence, jako například řešení problémů. Tyto zdroje však mohou dobře posloužit učitelům při prvotním seznámení s programem Scratch.

Procházením portálu scratched.media.mit.edu jsem objevil dva materiály, které jsou si postupem výuky podobné a to [12] a [15]. V obou zdrojích je patrné, že do výuky zařazují postupně všechny důležité prvky programování. Také na konci obsahují složitější projekt, který musí žáci řešit. Tím se také odlišují od ostatních mnou uváděných zdrojů. Mezi oběma zdroji je však také jeden zásadní rozdíl a to kdy zařazují do výuky proměnné. V případě [12] je to nesmyslně ve čtvrté lekci u příkladu, který použití proměnných nevyžaduje. V tomto případě autoři [15] zastávají názor, že téma proměnných patří ve Scratch k nejsložitějším a zařazují je až na konec výuky.

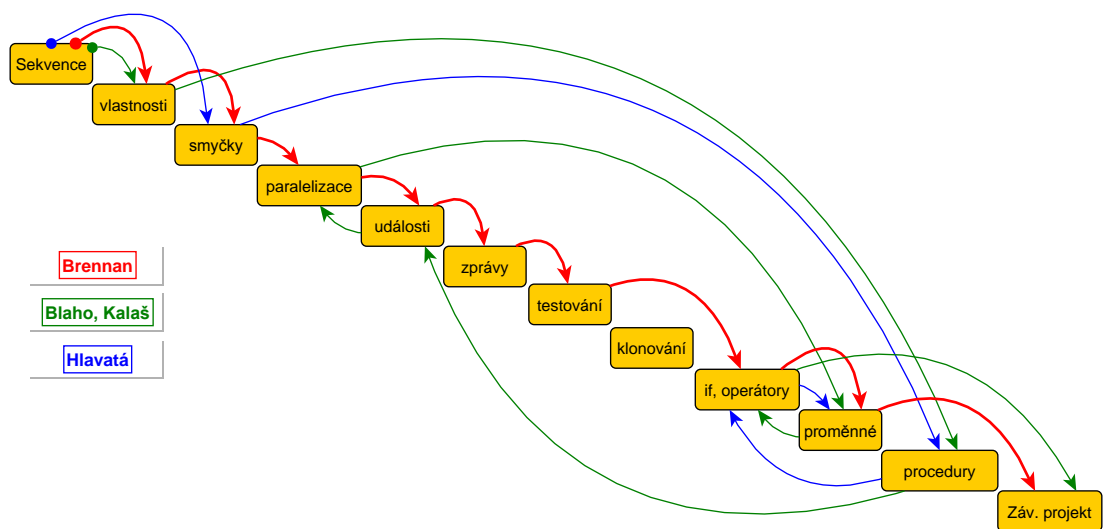
Porovnání postupů výuky Scratch



Obrázek 4.1: Porovnání postupu výuky Scratch

Na obrázku 4.1 jsem graficky znázornil postupy výuky, které používají učitelé v prostředí Scratch. Bloky jednotlivých témat představují mnou navrhovaný postup výuky. Na obrázku je dobře vidět, že ne všechny postupy výuky, které jsem výše analyzoval, postihují všechna témata. Téma klonování a procedury jsou novinkou ve verzi Scratch 2.0, proto se jim nevěnuje žádný autor.

Porovnání postupu výuku Scratch s postupy výuky pro Imagine a Baltika



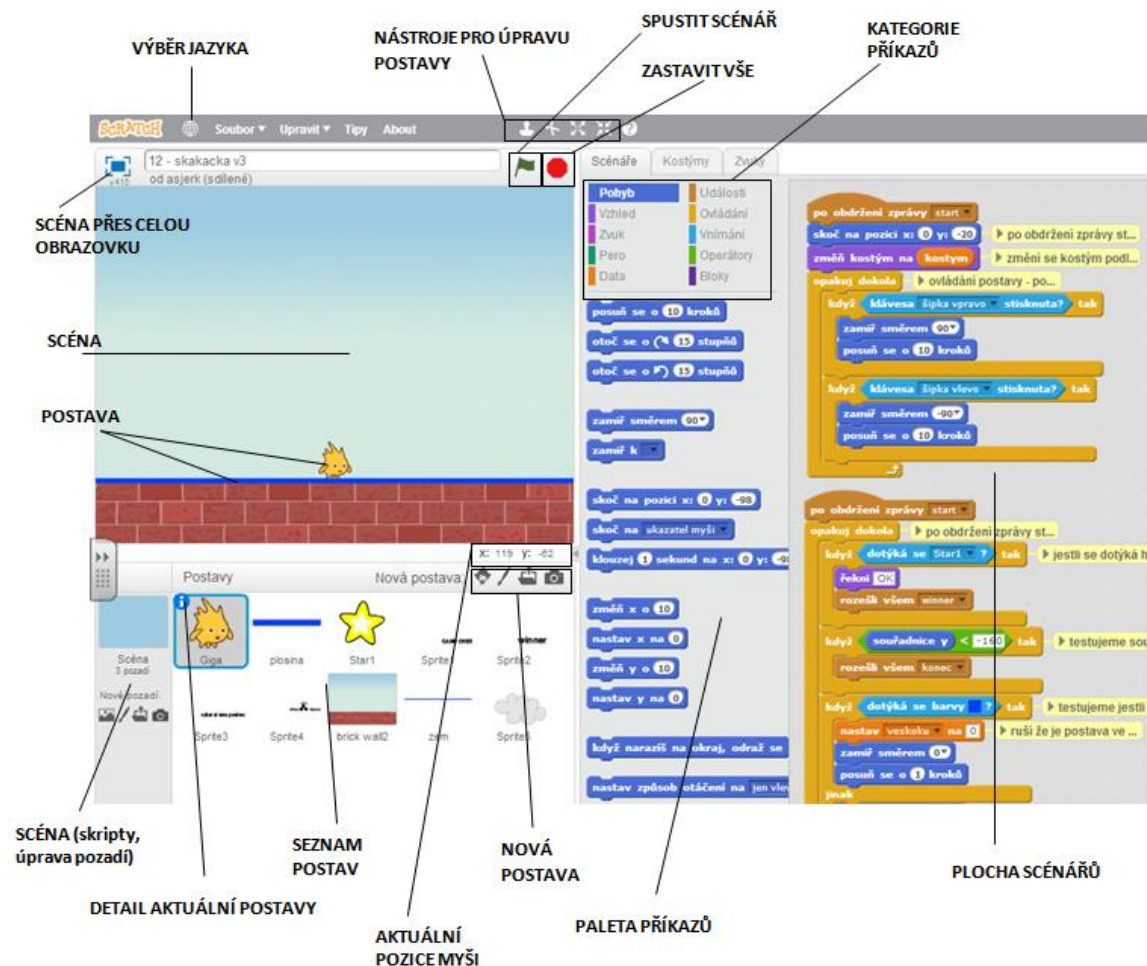
Obrázek 4.2: Porovnání postupu výuky Scratch, Imagine, Baltik

Obrázek 4.2 znázorňuje postupy výuku v prostředí Scratch s prostředími Imagine a Baltik. Bloky jednotlivých témat opět představují mnou navrhovaný postup výuky.

5. Programování ve Scratch 2.0

5.1 Seznámení s prostředím editoru Scratch

Prostředí Scratch, jak je možné vidět na obrázku 5.1, je rozděleno na tři hlavní části. V levé horní části je náhled výsledné scény, kde můžeme přímo sledovat, jak postavy reagují na námi připravený scénář. Levá dolní část obsahuje scénu a postavy. V této části můžeme změnit či vytvořit nové pozadí, vkládat nové postavy nebo postavám změnit vlastnosti. V pravé části okna se nachází editor programovacího projektu, tedy v závislosti na zvolené záložce jsou zobrazeny scénáře, kostýmy (případně pozadí u scény) a zvuky.



Obrázek 5.1: Prostředí Scratch

V prostředí scratch přetahujeme jednotlivé příkazy do plochy na pravé straně a skládáme je do sebe podobně jako puzzle. Tím je zajištěna maximální jednoduchost a není možné udělat syntaktickou chybu. Příkazy, které na ploše skládáme do sebe, tvoří takzvaný scénář. Postava (případně scéna) může mít více scénářů, které v závislosti na



nějaké události vykonává. Scénář vytváříme pro označenou postavu, případně scénu. Na obrázku 5.1 je zobrazen scénář pro postavu Giga.

Velkým specifikem a výbornou vlastností prostředí Scratch je názornost použití příkazů, především cyklů a podmíněného příkazu. Z vlastní zkušenosti vím, že někteří žáci mají problém pochopit používání cyklů i podmíněných příkazů u klasického programování. Bývá jim nejasné, které příkazy se vlastně opakují a které ne. V tomto je Scratch výbornou učební pomůckou pro všechny žáky začínající s programováním.

Jak je možné vidět z předchozího obrázku, každý blok (příkaz) má specifický tvar. Například bloky z kategorie vnímání lze díky svému tvaru korektně vložit jen do podmínek jiných bloků. Podobně jsou na tom operátory, proměnné a některé další bloky.

Proměnné lze v prostředí Scratch vytvářet z kategorie příkazů Data kliknutím na tlačítko Vytvořit proměnnou. Proměnnou můžeme vytvořit pro všechny postavy, pro tuto postavu, nebo jako cloud proměnnou. Cloud proměnná je uložena na serveru a slouží pro uchování hodnoty i po ukončení programu, například rekordu ve hře. Standardní proměnné uchovávají hodnotu jen po dobu běhu programu. Z kategorie bloků Data můžeme také vytvořit takzvaný seznam. Z hlediska klasického programování lze seznam používat jako pole (příkladem využití seznamu je hra had dostupná z mého studia). Může být vytvořen pro jednu postavu, stejně jako pro více postav. Seznam nemůže být uložen na serveru jako cloud proměnná.

V prostředí Scratch existují procedury (funkce) a nazývají se Bloky. Lze je vytvořit v kategorii příkazů Bloky tlačítkem „Vytvořit blok“. Nejprve je potřeba blok pojmenovat a přes volby k němu můžeme přidat parametry. Potvrzením nám vznikne

v oblasti scénářů blok . Pod tento blok vytvoříme určitou část programu (sled příkazů), které pak máme možnost kdykoliv zavolat ve scénářích této postavy. Volání procedury provedeme pomocí příkazu . Blok vytvoříme pro konkrétní postavu, nebo scénu a není dostupný pro scénáře jiných postav. Pokud chceme blok odstranit ze scénáře, je potřeba nejprve odstranit jeho volání.

5.2 Vzorový projekt

Specifika programování v prostředí Scratch bych chtěl demonstrovat na vzorovém projektu. Pro tento účel jsem vybral jednoduchou hru „Jumping to star“, kde cílem je skákat po náhodně generovaných plošinách až nahoru a dotknout se hvězdy. Celý projekt je dostupný online a můžete si jej zobrazit z mého studia <http://scratch.mit.edu/studios/202382/>, nebo přímo kliknutím na odkaz <http://scratch.mit.edu/projects/21257355/>. Projekt obsahuje většinu prvků programování (cykly, události, paralelně vykonávané scénáře, posílání zpráv mezi objekty, klonování, podmíněné příkazy, proměnné i bloky), které bychom chtěli žáky po dobu výuky postupně naučit.



Obrázek 5.2: Vzorový projekt Jumping to star

Koncepce hry

Z úvodní obrazovky si hráč kliknutím vybere postavu, se kterou bude hrát. Změní se pozadí a spustí se hra.

Nejprve je vygenerována spodní plošina (pevná zem), na které postava Giga stojí a je pro ni výchozí pozicí. Dále se začnou generovat plošinky, každá na náhodných pozicích osy x a vždy o 10 kroků výše na ose y. Zároveň se všechny objekty (pozadí i plošiny) začnou posouvat směrem dolů. Po vygenerování všech plošin se zobrazí v horní části obrazovky hvězda.

Postava po stisku klávesy šipka nahoru poskočí o několik kroků nahoru a pak bude padat dolů. Když postava Giga stojí na modré plošině, padá stejně rychle jako

plošina, jinak padá rychleji. Pomocí kláves šipka doleva a šipka doprava je možné posouvat postavu doleva a doprava.

Pokud postava spadne dolů – klesne pod hranici souřadnic osy y, která je nastavena na hodnotu -160 , je hra ukončena a zobrazí se postava s názvem konec hry. Pokud se postava dotkne hvězdy v horní části scény – hra je ukončena a zobrazí se postava s názvem vítězství.

Postava Giga

Postava Giga obsahuje pět scénářů, které budou postupně popsány níže.

Úvodní menu hry – výběr postavy



Obrázek 5.3: Vzorový projekt – menu hry

Úvodní obrazovka (obrázek 5.3) vzorového projektu má nastavené jiné pozadí, než to které bude použito ve hře. Obsahuje výběr postavy, se kterou chce hráč absolvovat hru. Menu je vytvořené pomocí postavy Giga, která má tři kostýmy. V každém ze tří opakování jen postavě nejprve změněn kostým a následně je postava naklonována na jiné místo ve scéně. Klouby reaguje na stisknutí myši a kurzor myši, kterého se dotýká. Tím je zvolen kostým, který se použije pro postavu ve hře.



Obrázek 5.4: Vzorový projekt – úvodní výběr postavy

Spuštění celého programu je zajištěno pomocí události *po kliknutí na zelenou vlajku*. Příkaz *opakuji* obsahuje změnu souřadnice x, výběr kostýmu a samotné klonování postavy. Činnost klonu je naprogramována v samostatném scénáři, jak ukazuje obrázek 5.4. Scénář klonu obsahuje podmíněný příkaz *když*. Podmínky jsou vytvořeny pomocí příkazů z kategorie vnímání. Zvolený klon nastaví proměnnou *kostym* na číselnou hodnotu svého kostýmu. Pro spuštění hry je využito události *rozešli všem* (zpráva), která je rozeslána všem postavám. Na zprávu zareagují ostatní postavy, případně scénáře, které začínají událostí *po obdržení zprávy*. Zprávy se v prostředí Scratch používají pro komunikaci mezi objekty (mezi postavami navzájem, mezi postavami a scénou).

Pohyb postavy

Následující obrázek 5.3 ukazuje další dva scénáře postavy Giga. První scénář se začne provádět ihned po obdržení zprávy *start*, která přijde od zvoleného klonu. Scénář běží paralelně se scénáři, které také zahájili svou činnost po obdržení zprávy *start*. Tento první scénář z obrázku 5.3 zajistí změnu kostýmu na základě hodnoty v proměnné *kostym*. Dále testuje stisknutí kláves *šipka vpravo* a *šipka vlevo*. V návaznosti na stisknutou klávesu postava půjde nebo poletí zvoleným směrem.



Obrázek 5.5: Vzorový projekt – postava Giga – pohyb postavy

Druhý scénář na obrázku 5.5 je zahájen vždy stisknutím klávesy *šipka nahoru*. Zajišťuje posun postavy směrem nahoru (skok). Používá proměnnou, která je dočasně nastavena na hodnotu 1, aby bylo zamezeno opakovanému skoku.

Testování podmínek



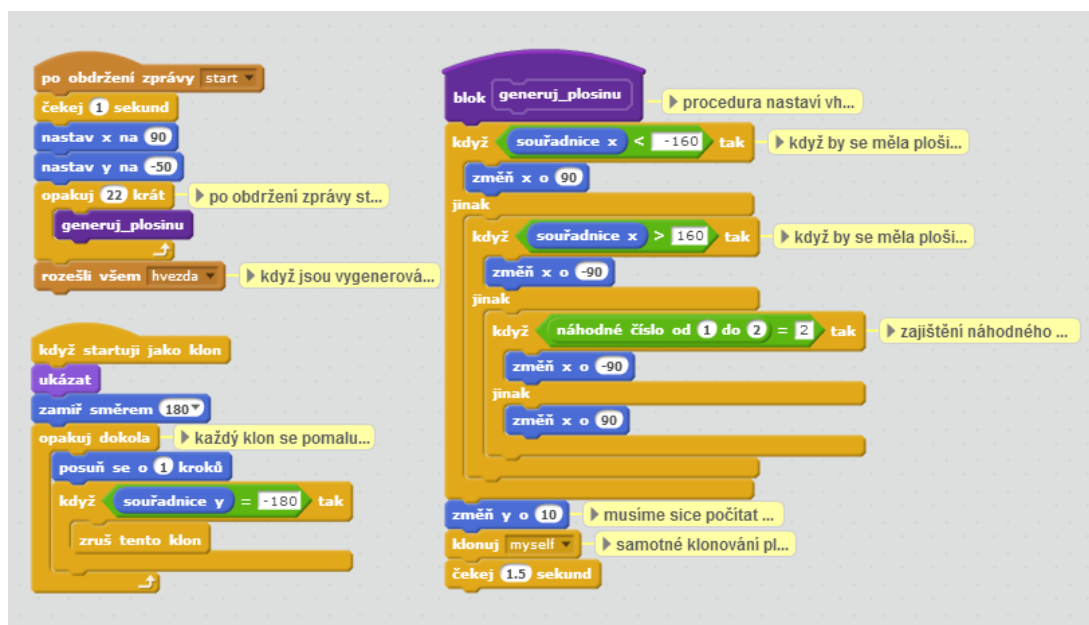
Obrázek 5.6: Vzorový projekt – postava Giga – testování podmínek

Scénář, který vidíte na obrázku 5.6, testuje pomocí podmíněného příkazu a příkazů vnímání, jestli se postava dotýká předmětů (plošinek, nebo cílové hvězdy). Pokud se nedotýká, zajišťuje scénář pád postavy. V případě, že se postava Giga dotkne postavy Star1, rozešle všem zprávu s názvem *winner*. Na základě přijetí zprávy může

zahájit svůj scénář jiná postava, konkrétně Sprite2. Pokud by postava Giga „spadla“ až pod souřadnice – 160 osy y, potom bude rozeslána zpráva s názvem *konec*.

Postava plošina

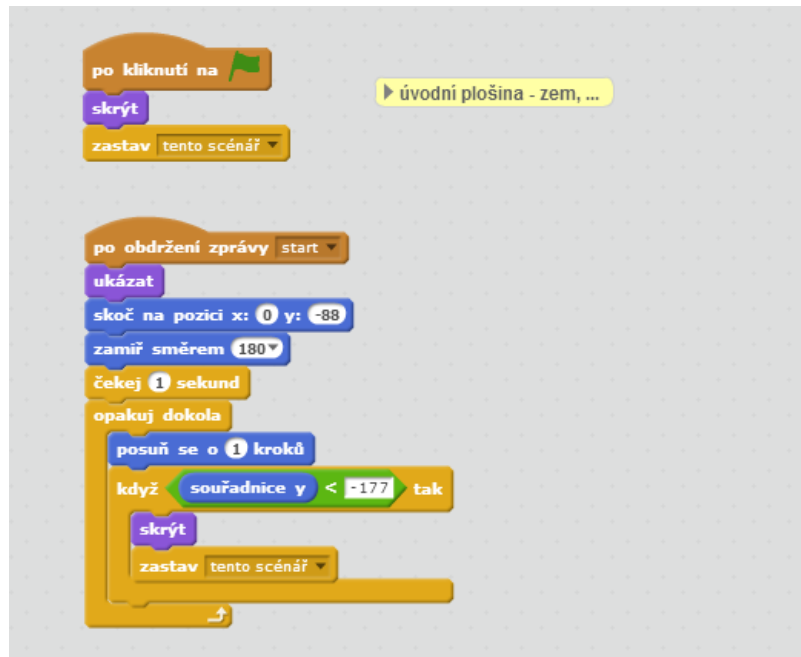
Další důležitá postava ve vzorovém projektu se nazývá *plošina*, její scénáře můžete vidět na obrázku 5.5. Postava klonuje sama sebe a tím vytváří záchytné body (plošiny) pro postavu Giga. Ve zdrojovém kódu je naprogramován blok, který je následně volán scénářem, který můžete vidět vlevo na obrázku 5.7. Blok *generuj plošinu* zjišťuje aktuální souřadnice osy x a podle toho nastavuje náhodně pozici pro generování plošiny doprava nebo doleva. V případě, že je plošina blízko okraje scény, generuje se nová plošina na druhou stranu od okraje. Generovaný klon se posouvá směrem dolů a při dosažení dolního okraje zmizí (klon je zrušen).



Obrázek 5.7: Vzorový projekt – postava plošina – náhodné generování padajících plošin

Postava zem

Tato postava je výchozí pozicí (pevnou zemí) pro postavu Giga. Zobrazí se na začátku hry, po obdržení zprávy *start*. Pomocí cyklu *opakuj dokola* se postava posouvá směrem dolů vždy o jeden krok. Jakmile dosáhne dolního okraje obrazovky, schová se. Testování souřadnice osy y je zajištěno pomocí příkazu *když* (obrázek 5.8).



Obrázek 5.8: Vzorový projekt – scénář postavy zem

Postava Star1



Obrázek 5.9: Vzorový projekt – postava Star1 je cílem hry

Tato postava je cíl hry, jehož se hráč snaží dosáhnout. Po spuštění programu je skrytá. Ukáže se po obdržení zprávy *hvezda* od postavy *plošina*. Pohyb postavy pomocí příkazu *klouzej* jen dokresluje scénu stoupání vzhůru.



Obrázek 5.10: Vzorový projekt – scénář postavy Star1

Postava posun pozadí

Tato postava nahrazuje pozadí z počátku hry. Je to z toho důvodu, abychom navodili dojem stoupaní vzhůru. Postava se pomalu posunuje směrem dolů. Pozadím nelze posunovat, ale postavou ano.



Obrázek 5.11: Vzorový projekt – scénář postavy „posun pozadí“

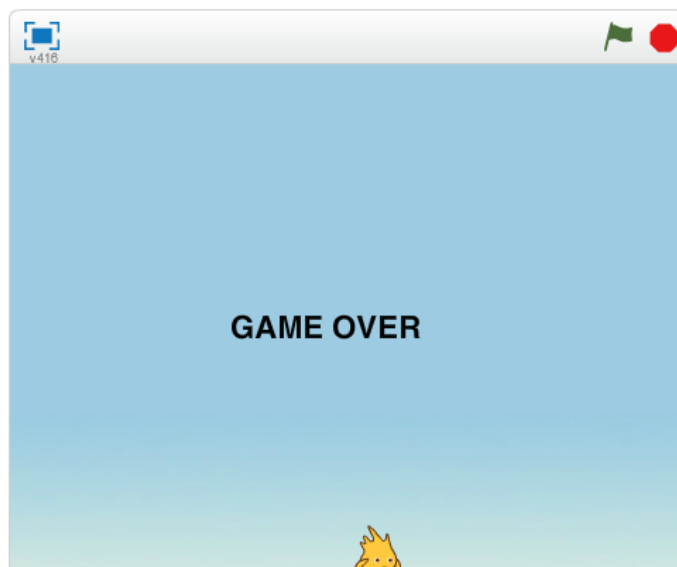
Postava obláček

Opět se jedná o postavu, která pouze dokresluje scénu stoupaní vzhůru. Obláčky jsou klonovány – náhodně se objevují a posunují se směrem dolů.



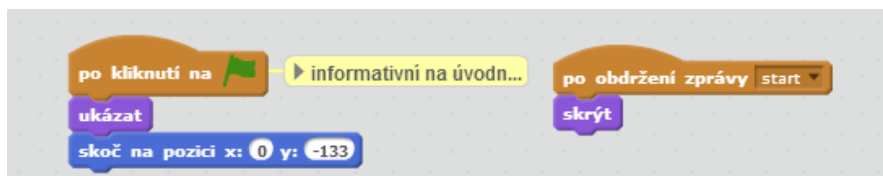
Obrázek 5.12: Vzorový projekt – scénář postavy obláček

Informativní postavy



Obrázek 5.13: Vzorový projekt – konec hry

Postavy „konec hry“ (obrázek 5.13), „vítězství“, „informace“ a „ovládání“ mají pouze informativní účel. Jsou to obrázky s různými texty. Všechny tyto postavy mají velmi podobné scénáře. Na základě nějaké události se jen skryjí nebo ukáží. Obrázek 5.14 ukazuje scénář jedné z postav.



Obrázek 5.14: Vzorový projekt – scénář postavy „ovládání“

6. Metodika výuky

Na základě prostudovaných zdrojů a analýzy výukových materiálů jsem sestavil vlastní metodiku výuky v prostředí programu Scratch. Navrhl jsem vlastní postup, který je přehledně shrnut v osnově výuky. Připravil jsem různé úlohy, které slouží k výuce a procvičování jednotlivých témat. K těmto úlohám jsou připravené metodické listy a žákovské listy. Metodický list obsahuje zadání pro žáky, obrázek projektu a metodické poznámky pro učitele. Žákovský list obsahuje zadání úlohy, obrázek výsledného projektu a v některých případech také rozpracovaný projekt.

6.1 Osnova výuky

1. Představení programu Scratch a seznámení s jeho prostředím
2. Posloupnost příkazů, cyklus s daným počtem opakování
3. Objekty (postavy a pozadí scény)
4. Více objektů (postav), postupný vývoj a testování
5. Paralelizace, události, nekonečný cyklus
6. Události, posílání zpráv mezi objekty
7. Testování a ladění
8. Klonování
9. Podmíněný příkaz a operátory
10. Proměnné, procvičení podmíněného příkazu a operátorů
11. Bloky (Procedury)
12. Závěrečný projekt

Odůvodnění zvoleného postupu výuky

Představení programu Scratch a seznámení s jeho prostředím

Prvním tématem je úplný úvod a seznámení s prostředím Scratch. Téma jsem zařadil na úplný začátek z několika důvodů. Žáci lépe poznají neznámý program a udělají si představu o jeho možnostech. Na začátku se žáci zaregistrují na portálu scratch.mit.edu.

Posloupnost příkazů, cyklus s daným počtem opakování

Zařazení posloupnosti příkazů je nezbytné v samém začátku. Při sestavování scénářů velmi brzy přijde potřeba opakovat příkazy. Proto je do druhého tématu zařazen

také příkaz opakování. Na rozdíl od běžných programovacích jazyků, kde se cykly zavádí do výuky později, je v programu Scratch použití cyklů jednoduché, intuitivní a velmi názorné. Proto je možné zařadit toto téma na začátek výuky.

Objekty (postavy a pozadí scény)

Třetí téma se zabývá objekty, jejich změnou a vlastnostmi. Důvodem zařazení na začátek výuky je jednoduchost probíraného tématu a potřeba využívat později změnu pozadí, kostýmů, nastavování vlastností objektům.

Více objektů (postav), postupný vývoj a testování

Jako jedna z důležitých dovedností u programování je postupný vývoj. Proto je toto téma zařazeno jako jedno z prvních témat. Scratch k tomuto přímo vybízí. Žáci naprogramují část scénáře a mohou ihned otestovat, jestli postava pracuje podle jejich představ. V rámci tohoto tématu se žáci naučí další dovednosti. Pro usnadnění pozdější práce potřebují umět kopírovat scénáře nebo příkazy a pracovat s více postavami.

Paralelizace, události, nekonečný cyklus

Dalším nezbytným prvkem programu Scratch jsou události, které žáci potřebují například ke spuštění naprogramovaného projektu. S tím také souvisejí paralelní činnosti postav. Témata jsou jednoduchá na pochopení a budou později běžně využívána, proto jsou zařazena do této části osnovy.

Události, posílání zpráv mezi objekty

Šesté téma je zaměřeno také na události, proto navazuje na předchozí část. V rámci tématu jsou také probírány zprávy. Jedná se o událost, díky které spolu objekty mohou komunikovat.

Testování a ladění

Testování a ladění je v programování důležitou činností, proto jsem je zařadil do osnovy před obtížnější témata. Pro žáky bude také velice přínosné, procvičit si orientaci ve zdrojovém kódu, který sami nevytvořili.

Klonování

Klonování je zařazeno jako osmé téma. Z hlediska náročnosti vytvoření scénáře pro klonovanou postavu se jedná o obtížnější téma, než která byla doposud probírána.

Podmíněný příkaz a operátory

Podmíněný příkaz je jeden z nejnáročnějších v prostředí Scratch a proto je zařazen až ke konci výuky, kdy žáci už umějí v prostředí velmi dobře pracovat. S ním souvisejí

také příkazy vnímání. Opět je tady rozdíl oproti klasickému programování, kde se podmíněné příkazy zařazují o mnoho dříve.

Proměnné, procvičení podmíněného příkazu a operátorů

Proměnné jsou v prostředí Scratch pokročilým tématem. Z počátku výuky žáci nepotřebují ve svých scénářích používat proměnné. Naproti tomu jsou v klasickém programování proměnné zařazovány na začátek výuky, ještě před podmíněné příkazy.

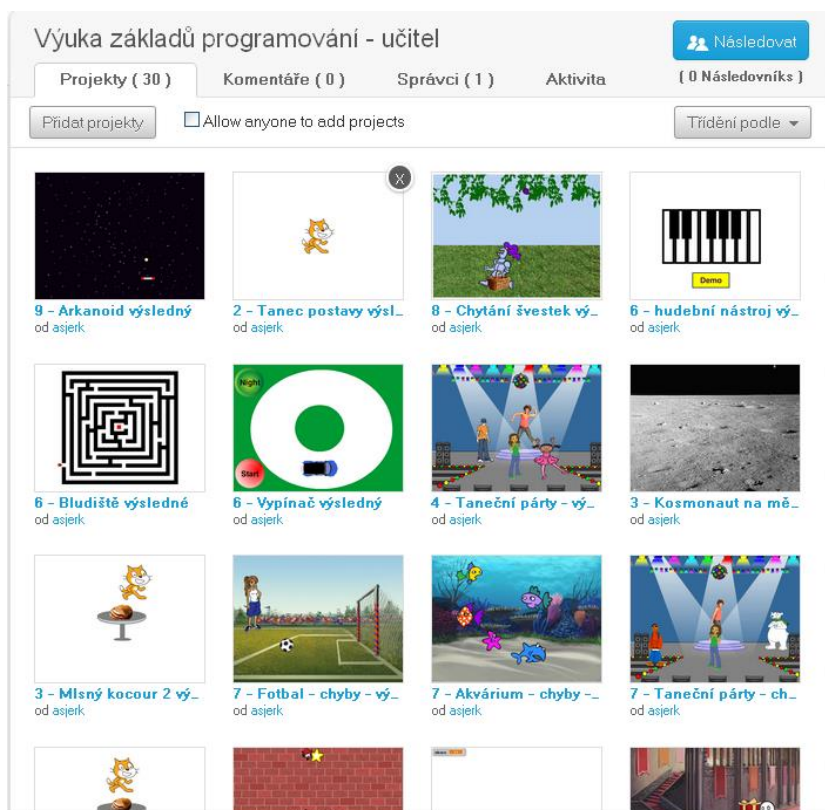
Bloky (Procedury)

Bloky (procedury a funkce) jsou novinkou v prostředí Scratch 2.0. Záměrně jsem je zařadil až na konec výuky. Jedná se opět o pokročilé téma, které při nedostatečné časové dotaci může být učiteli vynecháno. Žáci zvládnou naprogramovat libovolné projekty i bez pomoci bloků. Nicméně bloky mají význam z hlediska přehlednosti zdrojového kódu a opakovaného použití. Neměli by proto být záměrně opomíjené.

6.2 Postup výuky

Tato kapitola je pomůckou pro učitele při výuce základů programování v prostředí Scratch. Obsahuje cíle výuky, přehled aktivit a metodické poznámky. Jednotlivé téma připravené osnovy představuje vždy jednu podkapitolu postupu výuky. Ke každému tématu je také připraven jeden pracovní list pro žáky, který je přiložen jako příloha. Číslo a téma podkapitoly vždy koresponduje s připraveným pracovním listem a projektem ve studiu. Všechny programy, které zde jsou zmíněné, naleznete ve studiu s názvem „**Výuka základů programování – učitel**“ dostupný na adrese <http://scratch.mit.edu/studios/202382/>. Podobné studio je dostupné pro žáky, ale obsahuje pouze připravené projekty, které mají za úkol upravovat. Žákovské studio naleznete na adrese <http://scratch.mit.edu/studios/408206/>. Z důvodu, abychom žákům znesnadnili přístup k výsledným projektům, jsou připravené (rozpracované) projekty pouze v žákovském studiu a nejsou k dispozici v učitelově studiu.

Projekty umístěné ve studiu jsou pojmenované číslem (určuje téma, ke kterému projekt patří) a názvem úlohy. Například kapitola 5 obsahuje úlohu Kouzelná hůlka, proto výsledný projekt pro učitele se jmenuje: 5 – Kouzelná hůlka výsledný.



Obrázek 6.1: Studio připravené pro učitele

6.2.1 Seznámení s prostředím Scratch

Cíl výuky

- Žák aktivně pracuje s webem programu Scratch, prochází sdílené projekty a spouští je,
- dokáže se zaregistrovat na portálu a chápe důležitost vlastního účtu,
- spustí program Scratch 2.0, orientuje se v programu, dokáže vytvořit, uložit a sdílet svůj projekt,
- naprogramuje jednoduchý scénář objektu – postavy.

Aktivity

- Seznámení s webem <http://scratch.mit.edu/>
- Procházení dostupných projektů
- Registrace na portálu
- Seznámení s prostředím Scratch 2.0
- Programování neobyčejné postavy

Seznámení s webem Scratch

Výuku začneme tím, že žákům představíme webové stránky programu Scratch <http://scratch.mit.edu/>. Ukážeme některé vzorové projekty dostupné z tohoto portálu. Jako kontrolu žákům položíme otázku: „Kolik je aktuálně sdílených projektů na stránkách Scratch?“.

Procházení dostupných projektů

Žáky necháme prozkoumávat web a projít několik projektů. Dáme jim dostatek prostoru k tomu, aby si projekty spustili a pokud narazí na nějaké hry, mohou si je klidně zahrát. Tím získají motivaci pro vlastní práci se Scratchem. Důležité je také ukázat, jakým způsobem mohou pohlédnout dovnitř projektů.

Registrace na portálu

Seznámíme žáky s výhodami, které nám přináší registrace na portálu scratch.mit.edu a ukážeme jak registraci provést. Dáme žákům prostor, aby se zaregistrovali.

Seznámení s prostředím Scratch 2.0

Ukážeme žákům, jak lze spustit prostředí Scratch 2.0. V krátkosti je seznámíme s jeho uspořádáním. Upozorníme žáky na cloudové řešení prostředí a s tím

související možnosti, především na možnost spustit si Scratch 2.0 odkudkoliv kde máme přístup k internetu a zároveň mít k dispozici své projekty.

Programování neobyčejné postavy

Zadání pro žáky:

Nejprve si otevřete projekt s názvem **1 – První program**, který naleznete v připraveném studiu na adrese <http://scratch.mit.edu/studios/408206/>. Poté musíte stisknout tlačítko vpravo nahoře – **Pohlédni dovnitř**.

Tento program můžete otevřít také pomocí odkazu <http://scratch.mit.edu/projects/21230742/#editor>.

Použijte příkazy, které leží na ploše, na níž se programuje. Příkaz spustíte tak, že na něj kliknete levým tlačítkem myši. Klikejte na jednotlivé příkazy a pozorujte nebo poslouchajte postavu v levé horní části okna.

Měňte číselné hodnoty u příkazů a pozorujte reakce postavy, když příkaz spustíte.

Na plochu můžete přetahovat další příkazy a vyzkoušet si je.

Příkazy můžete skládat do sebe podobně jako puzzle a vytvářet tak scénář. Takto složený scénář můžete spustit kliknutím, stejně jako jednotlivé příkazy.

O nové poznatky a příkazy se podělte s ostatními spolužáky.



Obrázek 6.2: Projekt 1 – První program

Poznámky pro učitele:

Když žáky seznámíme s prostředím editoru Scratch 2.0, necháme je otevřít připravený projekt **1 – První program**. Na ploše, na které se programuje, jsou připraveny některé příkazy. Žáci si je mají možnost spouštět kliknutím na ně a pozorovat reakce postavy. Žáci po otestování příkazů z plochy budou přetahovat další příkazy na plochu a také skládat a spouštět svůj jednoduchý scénář.

6.2.2 Posloupnost příkazů a cyklus

Cíl výuky

- Žák popíše přesný postup libovolné činnosti (algoritmus),
- vytvoří scénář postavy, kde budou za sebou uspořádané jednotlivé příkazy (posloupnost příkazů),
- použije příkaz *opakuj* pro opakování příkazů,
- chápe význam cyklů ve svém scénáři.

Aktivity

- Diskuze s žáky – přesný postup
- Tanec bez PC
- Tanec postavy
- Mlsný kocour

Diskuze s žáky

Zadání pro žáky:

Promyslete a následně vyjmenujte 5 situací ze života, ve kterých budete postupovat podle přesně definovaných kroků (instrukcí, pokynů nebo příkazů).

Poznámky pro učitele:

Výukou chceme docílit toho, aby se žáci zamysleli a uvědomili si, že přesný postup činnosti se vyskytuje v běžném životě všude kolem nás. Proto vyvoláme diskuzi (nebo brainstorming), do které bychom chtěli, aby se zapojili všichni žáci a sdělovali své nápady. Budeme chtít od nich jmenovat různé návody a postupy, například kuchařský recept, postup výroby, matematický postup a další. Tuto myšlenku potom musíme spojit s programováním.

Tanec bez PC

Zadání pro žáky:

Vytvořte dvojice. Jeden ze dvojice bude dávat instrukce druhému a ten je bude provádět. Jedná se o jednoduchý tanec a pokyny znějí následovně:

„Doprava - krok, krok, tleskni, doleva – krok, krok, tleskni.“

Tyto instrukce opakujte a kamarád bude tancovat.

Instrukce, které udělíte kamarádovi, libovolně pozměňte a přidejte další.

Poznámky pro učitele:

Cvičení „Tanec“ je zaměřeno na pochopení posloupnosti příkazů a jejich opakování. Ze třídy vybereme několik dvojic, které postupně tříde předvedou tanec. Žákům rozdělíme úlohy, jeden žák bude udílet příkazy a druhý je bude provádět. Důraz musíme klást na to, aby žák dělal jen povely, které mu kamarád říká. Žáci si tím uvědomí, že program provede pouze ty příkazy, které mu napíšeme a to v přesném pořadí za sebou. Z této situace musí také vyplynout, že dané posloupnosti příkazů se opakují.

Tato aktivita vychází z materiálu Scratch curriculum guide od Karen Brennan [15].

Tanec postavy

Zadání pro žáky:

Nejprve si otevřete projekt **Tanec postavy**, který naleznete v připraveném studiu na adrese <http://scratch.mit.edu/studios/408206/>.

Na základě předchozího úkolu (Tanec bez PC) nyní naučte stejný tanec postavu ve Scratchi. Postavě kočky nebo jiné udílejte povely tak, aby postava tančila ve vaší scéně. K tomu použijte příkazy, které leží na ploše, na níž se programuje.

Přijďte na to, jak taneční pohyb postavy 10krát zopakovat?

Až budete mít projekt hotov, můžete se s ním pochlubit ostatním například tak, že jej budete sdílet. Ostatní lidé jej budou moci prohlížet přes webové stránky scratch.mit.edu.

Poznámky pro učitele:

Žáky nyní pobídneme k naprogramování tance kočky (nebo libovolné postavy) v programu Scratch. Žáci využijí příkazy ležící na ploše a sestaví je do sebe. Tím vznikne určitá posloupnost příkazů – blok příkazů, na který když žáci kliknou, uvidí ve scéně výsledek své práce – pohyb postavy.

Opakovaným klikáním docílí toho, že bude kočka tančit. Necháme žáky, jestli přijdou na to jak příkazy opakovat. Případně je zkusíme pobídnout otázkou, jestli takový příkaz naleznou.

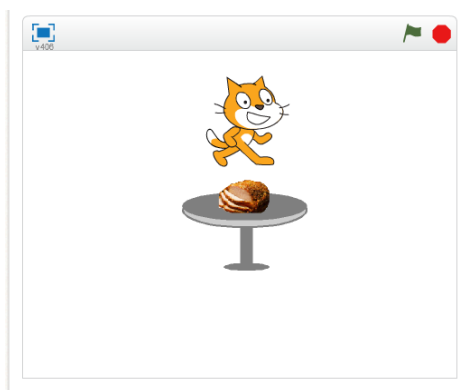
Jako ukázkou a představu rozsahu programu si zobrazte projekt **Tanec postavy** výsledný z připraveného studia pro učitele.

Tato aktivita vychází z materiálu Scratch curriculum guide od Karen Brennan [15].

Mlsný kocour

Zadání pro žáky:

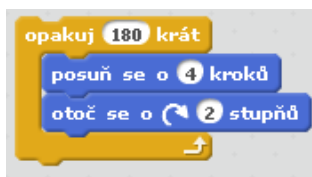
Využijte připraveného projektu „Mlsný kocour“ k naprogramování kocoura, který hladově obchází okolo stolu s pečinkou. Projekt je dostupný z připraveného studia <http://scratch.mit.edu/studios/408206/>. Projekt převezměte a podle svého upravte. Proto, aby bylo možné projekt upravovat, nejprve musíte pohlédnout dovnitř a stisknout oranžové tlačítko umístěné vpravo nahoře „Odvozené“. Tím si projekt zkopírujete a uložíte k sobě do svých věcí a můžete na něm začít pracovat.



Obrázek 6.3: Mlsný kocour

Poznámky pro učitele:

Žáci otevřou připravený projekt s názvem Mlsný kocour. Chtěli bychom, aby si projekt převzali (odvodili). Je pravděpodobné, že žáci naprogramují pohyb kolem stolu pomocí vícenásobného klouzání, nebo vícenásobného použití příkazu posuň se. Naším cílem je, aby o tom přemýšleli a zkusili vymyslet co možná nejjednodušší řešení – **využili opakování**.



Obrázek 6.4: Mlsný kocour – příklad možného řešení

Jako ukázkou vhodného řešení si prosím zobrazte projekt „**Mlsný kocour výsledný**“, který je dostupný z připraveného studia pro učitele.

6.2.3 Postavy a pozadí

Cíl výuky

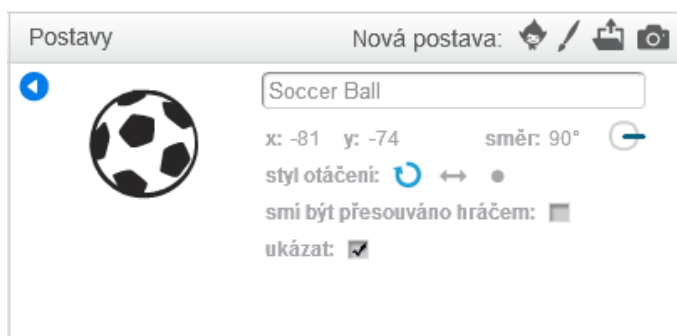
- Žák dokáže vytvořit postavu nebo pozadí (z knihovny, kreslením, importem, načtením z kamery),
- graficky upraví vzhled postavy nebo scény,
- upraví vlastnosti postavy nebo scény,
- umístí objekt na požadované místo scény, orientuje se v souřadnicovém systému scény,
- dokáže měnit vzhled postavy – například skrýt postavu nebo změnit efekt postavy.

Aktivity

- Ukázka možností změn postav a scény
- Kosmonaut na měsíci
- Pavouk
- Kosmonaut na měsíci 2
- Mlsný kocour 2

Ukázka možností změn postav a scény

Nejprve bychom žákům ukázali, jakými způsoby mohou změnit obrázek postavy a vytvořit novou postavu. Také bychom jim ukázali, že stejným způsobem to lze provést u pozadí. Ukážeme také, jak lze měnit vlastnosti postavy (obrázek 6.5), nebo jak to samé provést pomocí příkazů.



Obrázek 6.5: Vlastnosti postavy

Kosmonaut na měsíci

Zadání pro žáky:

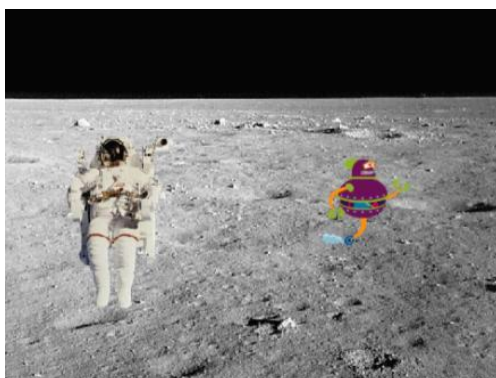
Otevřete si Scratch a pojmenujte prázdný projekt.

Pokuste se změnit pozadí tak, aby scéna připomínala prostředí na Měsíci nebo ve Vesmíru podobně jako ukazuje obrázek vpravo.

Dokážete postavu kočky změnit na jinou postavu (kosmonauta)? Najděte (například na internetu) vhodný obrázek, který využijete pro změnu.

Postavu pak naprogramujte, ať se vznáší ve Vesmíru.

Pokuste se přidat do vesmírného prostoru ještě další objekt, který by tam mohl létat.



Obrázek 6.6: Kosmonaut na měsíci

Poznámky pro učitele:

Úloha je zaměřena na změnu pozadí z knihovny obrázků a změnu postavy importem obrázku nalezeného na internetu. Žáci by v průběhu vypracování úlohy měli přijít na to, jak měnit vlastnosti postavy (obr. 6.5). Také si vyzkoušejí vložení další postavy. Seznamte žáky s událostí *po kliknutí na mě*. Tuto událost nyní mohou žáci využít, aby po kliknutí postava začala plnit povely ze svého scénáře. Dále je tato úloha zaměřena na procvičení posloupnosti příkazů a jednoduché smyčky. Jako ukázkou a představu rozsahu programu lze využít projekt **Kosmonaut na měsíci výsledný** ze studia připraveného pro učitele.

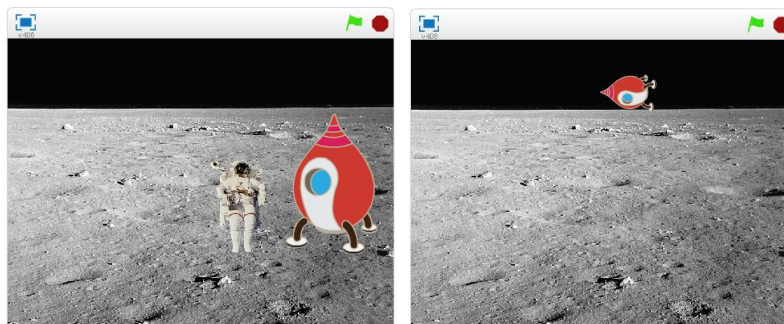
Kosmonaut na měsíci 2

Zadání pro žáky:

Využijte již vytvořený projekt s kosmonautem, uložte si jej jako kopii a upravte jej.

Vložte do scény postavu, která bude představovat raketu.

Scénář vytvořený pro kosmonauta upravte tak, aby doskákával k raketě a nastoupil do ní. Raketa s ním nakonec odletí.



Obrázky 6.7 a 6.8: Upravený projekt s kosmonautem a raketou

Poznámky pro učitele:

Žáci by v úloze měli použít příkaz pro skrytí postavy z kategorie *vzhled*. Měli bychom je směřovat na to, aby využili u postav vhodné časování a prozatím nepoužívali podmínky a vnímání. Dále bychom chtěli procvičit použití cyklů. Učitel zde může žáky navést k použití vnořeného cyklu (obr. 6.9). Příkladem možného řešení této úlohy je projekt **Kosmonaut na měsíci 2 výsledný**, dostupný ve studiu pro učitele.



Obrázek 6.9: Příklad správného (vlevo) a špatného řešení

Pavouk

Zadání pro žáky:

Vaším úkolem v této úloze je postavit s pavoukem pavučinu.

Z připraveného studia si můžete odvodit projekt s názvem Pavouk (pavouka můžete také namalovat, nebo importovat obrázek).

V připraveném projektu je pouze postava pavouka bez scénáře. Projekt prozatím neobsahuje ani pozadí.

Nejprve změňte pozadí, vyberte pro scénu vhodný obrázek.

Postavu pavouka naprogramujte tak, aby při svém pohybu stavěl pavučinu.

Může pavouk za sebou zanechávat stopu (pavoučí vlákno)?



Obrázek 6.10: Pavouk

Poznámky pro učitele:

Úloha je zaměřena na procvičení vlastností postavy a pozadí, procvičení posloupnosti příkazů a opakování. Žáci se také seznámí s použitím souřadnic ve scéně. V úloze žáci objeví nástroj pero a přijdou na to, že pomocí něho může pavouk tvořit své vlákno. Jako ukázkou a představu rozsahu programu lze využít projekty **Pavouk výsledný verze 1**, nebo jednodušší **Pavouk výsledný verze 2**.

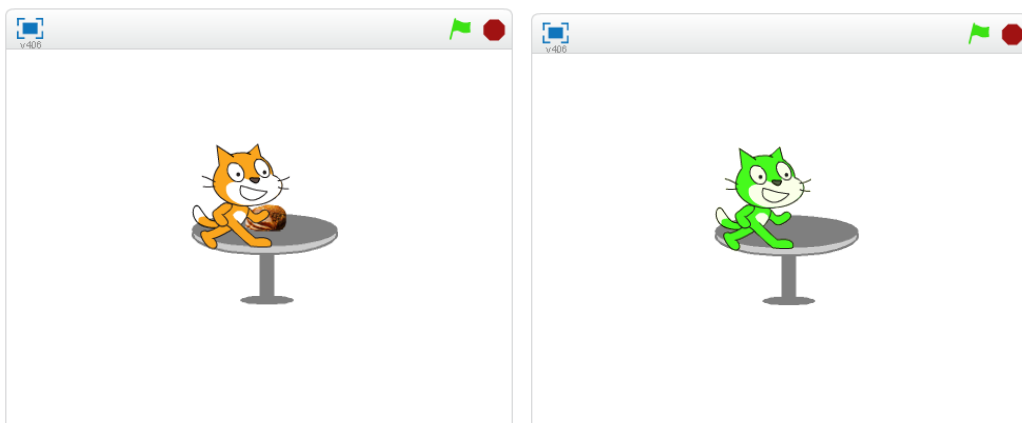
Mlsný kocour 2

Zadání pro žáky:

Úlohu „Mlsný kocour“ se pokuste upravit tak, aby kocour skočil na stůl a snědl pečínku.

Bude to i na něho dost velká porce a začne mu z toho být špatně. Dokážete ukázat, jak je z mastné pečínky kocourovi špatně?

Úpravy proveďte ve svém upraveném projektu nebo v připraveném projektu Mlsný kocour.



Obrázky 6.11 a 6.12: Upravený projekt Mlsný kocour 2

Poznámky pro učitele:

Žáci by v úloze měli použít příkazy z kategorie *vzhled* – různé efekty. Měli bychom je směřovat na to, aby využili u postav vhodné časování a prozatím nepoužívali podmínky a vnímání. Příkladem možného řešení této úlohy je projekt **Mlsný kocour 2 výsledný**, dostupný ve studiu pro učitele.



Obrázek 6.13: Mlsný kocour 2 – možné řešení scénářů postav

6.2.4 Více postav a postupný vývoj

Cíl výuky

- Žák chápe a používá při programování postupný (inkrementální) vývoj,
- vloží do scény více postav,
- každé postavě dokáže naprogramovat vlastní scénář,
- kopíruje scénáře mezi postavami,
- kopíruje postavy,
- vytvoří postavě více kostýmů a naprogramuje jejich změnu.

Aktivity

- Kontrola porozumění pojmů
- Taneční párty

Kontrola porozumění pojmů

Zeptejte se žáků:

- Jaký je rozdíl mezi postavou a kostýmem postavy?
- Kdy použijete postavu?
- Kdy použijete kostým?

Naším cílem je, aby žáci porozuměli pojmu postava a kostým, chápali rozdíl mezi nimi. Měli by umět použít novou postavu, když potřebují ve scéně další objekt. Naopak změnu kostýmů používat v případě, když potřebují měnit vzhled jedné postavy (například taneční pózy v úloze Taneční párty).

Taneční párty

Zadání pro žáky:

Otevřete si program Scratch a v něm nový projekt.

Upravte pozadí scény, použijte vhodný obrázek z knihovny, nebo nějaký vlastní.

Do scény také přidejte libovolné postavy.

Postavy naprogramujte tak, aby reagovaly na kliknutí tancem.

Dokážete na párty přehrávat svou oblíbenou píseň? Pokuste se vložit a přehrávat ve scéně vlastní oblíbenou píseň.

Pracujte postupně po malých částech, vytvořenou část programu vždy otestujte. Pokud bude vše procovat správně, podle vašich představ, pokračujte dál ve vytváření scénáře.

Pokud je to možné, nafoťte se v různých tanečních pózách. Fotografie použijte ve svém projektu tak, že budete tančit na tanečním parketu.

Své práce můžete na závěr sdílet s ostatními.



Obrázek 6.14: Taneční párty

Poznámky pro učitele:

Žáci vloží do scény více postav. Tanec by měli naprogramovat pomocí změny kostýmů. Žáky navedeme na použití události *po kliknutí na mě*, kterou použijí pro spuštění scénáře postavy. Během práce budeme žáky nabádat, aby pracovali postupně – vždy vytvořili malou část kódu a otestovali.

Jako ukázkou pro žáky nebo jen pro představu o rozsahu programu lze využít projekt **4 – Taneční párty výsledný** z připraveného studia pro učitele.

Na závěr vyberte některé pěkné práce a požádejte jejich autory, aby se podělili s ostatními o nové poznatky, případně ukázali zajímavou část programů. Žáci také mohou své projekty sdílet a tím je ukázat ostatním.

Tato aktivita vychází z materiálu Scratch curriculum guide od Karen Brennan [15].

6.2.5 Paralelizace a události

Cíl výuky

- Žák naprogramuje postavu pro vykonávání více scénářů v jeden okamžik,
- naprogramuje postavy tak, že pracují všechny současně,
- používá ve svých scénářích nekonečný cyklus,
- dokáže použít událost *po kliknutí na zelenou vlajku* jako centrální spouštěcí událost pro všechny objekty.

Aktivity

- Animace akvária
- Kouzelná hůlka

Animace akvária

Zadání pro žáky:

Otevřete si program Scratch a v něm nový projekt.

Upravte pozadí scény, použijte vhodný obrázek z knihovny, nebo nějaký vlastní.

Do scény také přidejte libovolné postavy (vodní živočichy).

Postavy naprogramujte tak, aby samy plavaly v akváriu.

Přijďte na to, jak projekt naprogramovat, aby se všechny postavy začaly pohybovat na jediný povel – například při kliknutí na zelenou vlajku?

Může se postava pohybovat neustále?

Co se stane, když narazí na okraj?

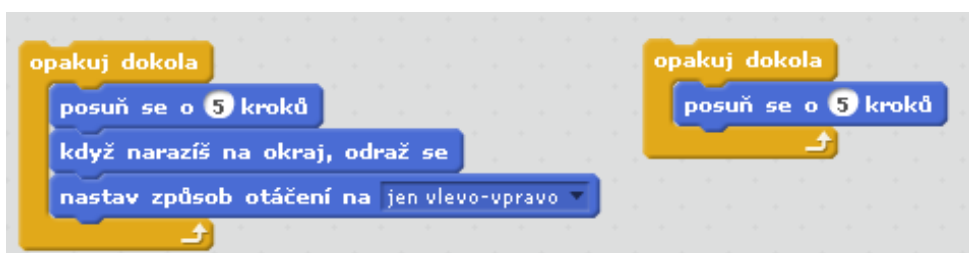
Lze zajistit, aby se postava při změně směru pohybu neotáčela (myšleno hlavou dolů)?



Obrázek 6.15: Akvárium

Poznámky pro učitele:

Žáci v této úloze naprogramují každé postavě nekonečný pohyb pomocí příkazu *opakuj dokola*, který bude spuštěn kliknutím na zelenou vlajku. Mohou si pomáhat tím, že budou kopírovat scénáře mezi postavami. Dále by měli být schopni použít příkaz na odrazení od okraje scény a upravit vlastnosti postavám (například styl otáčení). V této úloze mohou být žáci kreativní. Jako ukázkou pro žáky nebo jen pro představu o rozsahu programu lze využít projekt **5 – Animace akvária výsledný** z připraveného studia pro učitele.



Obrázek 6.16: Akvárium – příklad správného řešení (vlevo), typické chyby (vpravo)

Kouzelná hůlka

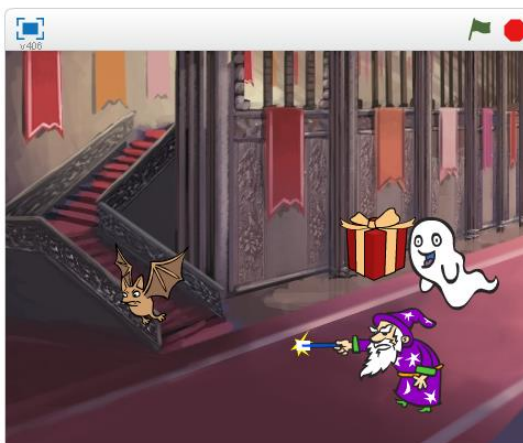
Zadání pro žáky:

Zahrajeme si na kouzelníka. Využijte připravený projekt s názvem **Kouzelná hůlka**, který naleznete v připraveném studiu <http://scratch.mit.edu/studios/408206/>.

Projekt obsahuje pozadí a několik různých postav.

Naprogramujte postavám pohyb, případně další efekty.

Připravenou kouzelnou hůlku naprogramujte tak, aby se pohybovala spolu s kurzorem myši. Kliknutím na postavy budete kouzlit a postavy měnit v něco úplně jiného, nebo je necháte zmizet.



Obrázek 6.17: Kouzelná hůlka

Poznámky pro učitele:

Žáci by měli naprogramovat pohyb u některých postav – využijí opět příkaz *opakuj dokola*. Postavy mohou mít více scénářů. Žáci by měli přijít na to, jak přichytit hůlku ke kurzoru myši (opakování + skok na kurzor myši). Také použijí události ke spuštění celé scény a ke změně kostýmů jednotlivých postav (kouzlení).

Na ukázkou pro žáky nebo pro představu si zobrazte projekt **5 - Kouzelná hůlka** – výsledná z připraveného studia pro učitele.



Obrázek 6.18: Postava kouzelna_hulka



Obrázek 6.19: Paralelní scénáře

6.2.6 Události a zprávy

Cíl výuky

- Žák dokáže použít ve svých scénářích událost „po kliknutí na mě“ a událost „po stisku klávesy“,
- naprogramuje objekt tak, aby rozeslal všem objektům zprávu,
- naprogramuje objekt tak, aby reagoval na přijatou zprávu vhodným scénářem.

Aktivity

- Hudební nástroj
- Bludiště
- Vypínač
- Razítka

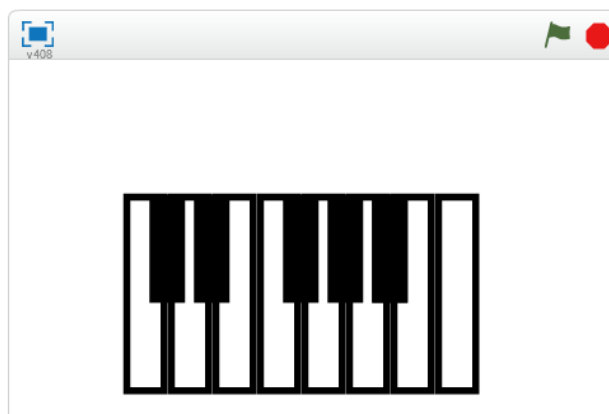
Hudební nástroj

Zadání pro žáky:

Otevřete si editor Scratch a v něm nový projekt.

Vytvořte a naprogramujte hudební nástroj (například klavír), na který bude možné interaktivně hrát (klikáním myši).

Na internetu, nebo v hodině hudební výchovy najděte jednoduchou píseň a pokuste se ji přehrát na vašem nově vytvořeném nástroji.



Obrázek 6.20: Hudební nástroj

Poznámky pro učitele:

Úloha je zaměřena především na událost *po kliknutí na mě*. Kliknutím na připravenou klávesu bude přehrán zvuk.

Úloha také předpokládá mezipředmětový vztah s hudební výchovou. Chtěli bychom, aby žáci našli nějakou jednoduchou píseň a tu zahráli na svém vytvořeném nástroji.

Na ukázkou pro žáky nebo pro představu rozsahu projektu si prosím zobrazte projekt **6 – Hudební nástroj – výsledný** z připraveného studia pro učitele.

Bludiště

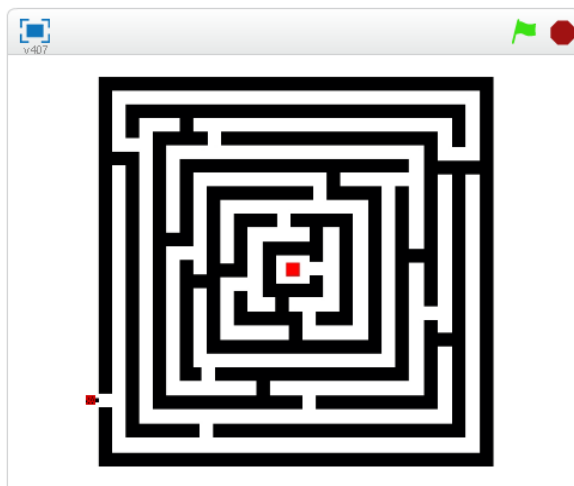
Zadání pro žáky:

Otevřete si projekt Bludiště z připraveného studia.

Úkolem je využít události k dokončení rozpracovaného projektu.

V projektu je připravené pozadí (bludiště) a postava, která musí bludištěm projít.

Naprogramujte postavu tak, aby její pohyb byl možný ovládat pomocí šipek na klávesnici.



Obrázek 6.21: Bludiště

Poznámky pro učitele:

Úloha je zaměřena především na událost *po stisku klávesy*. Stisknutím příslušné klávesy musí být postava posunována bludištěm.

Na ukázkou pro žáky nebo pro představu rozsahu projektu si prosím zobrazte projekt **6 – Bludiště – výsledné** z připraveného studia pro učitele.

Vypínač

Zadání pro žáky:

Otevřete si projekt **Vypínač** z připraveného studia.

Úkolem je využít události – zprávy k dokončení rozpracovaného projektu.

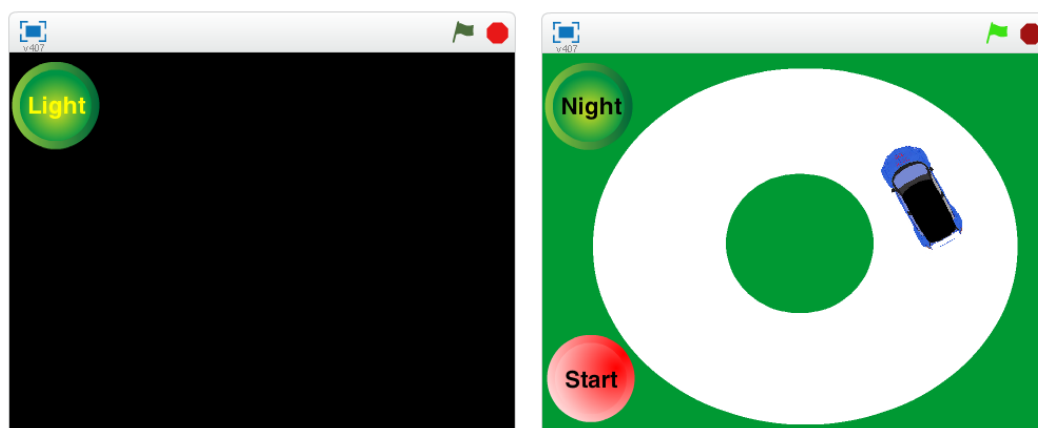
V projektu máte připravené pozadí, tlačítka i automobil.

Při spuštění projektu chceme, aby byla tma a bylo viditelné pouze tlačítko s nápisem Light. Po stisku tohoto tlačítka se nám rozsvítí světlo a ukáže se nám pozadí dráhy, automobil a tlačítko Start. Stiskem stejného tlačítka bychom měli mít možnost kdykoliv zhasnout světlo.

Stiskem tlačítka Start dostane automobil povel a začne se pohybovat po dráze.

K naprogramování komunikace mezi postavami navzájem a scénou využijte zprávy.

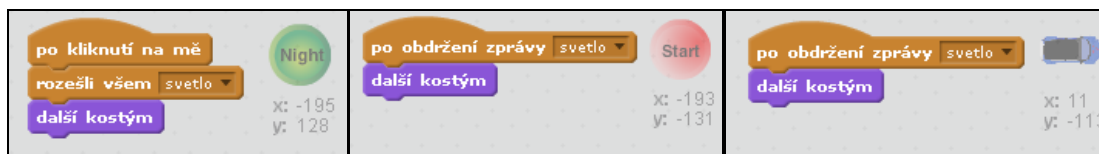
Každá postava má více kostýmů, které využijte pro dosažení nejlepšího výsledku.



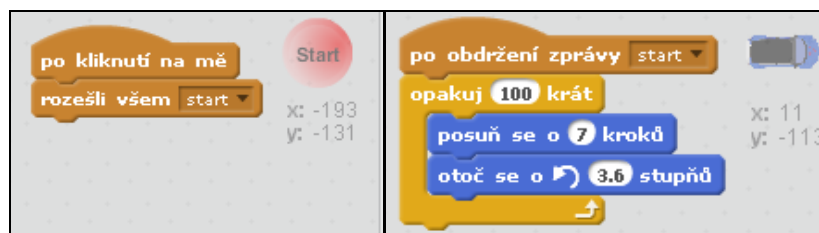
Obrázky 6.22 a 6.23: Projekt Vypínač

Poznámky pro učitele:

Úloha je zaměřena především na události posílání zpráv. Tlačítko odešle zprávu a ostatní objekty (scéna i postavy) musejí reagovat. Například když stiskneme tlačítko Light, rozešle všem zprávu. Scéna reaguje na základě přijaté zprávy změnou pozadí. Postavy auto a tlačítko Start zase reagují na přijatou zprávu změnou kostýmu.



Obrázek 6.24: Zpráva „svetlo“ u všech tří postav



Obrázek 6.25: Zpráva „start“ u postav „Tlačítko_start“ a „auto“

Na ukázkou pro žáky nebo pro představu rozsahu projektu si prosím zobrazte projekt **6 – Vypínač – výsledný** z připraveného studia pro učitele.

Razítka

Zadání pro žáky:

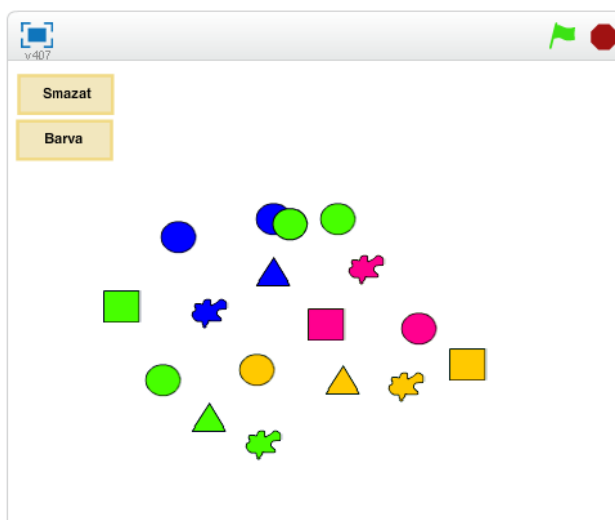
Otevřete si editor Scratch a v něm nový projekt.

Vaším úkolem je nakreslit jednoduchou postavu (razítka). Pro razítka vytvořte více tvarů (kostýmů), které bude možné přepínat například stisknutím klávesy mezerník.

Razítka bude přichycené k myši a na kliknutí se otiskne do scény.

Vytvořte tlačítko, pomocí kterého bude možné měnit barvy razítka.

Vytvořte ještě jedno tlačítko, které bude moci vymazat plochu.



Obrázek 6.26: Razítka

Poznámky pro učitele:

V této úloze můžeme nechat žáky tvořit libovolné tvary razítek (kostýmy). Pro spuštění scénářů by měli použít události (především *po kliknutí na mě*, *po stisku klávesy*) a posílání zpráv (u tlačítek).

Na ukázkou pro žáky nebo pro představu rozsahu projektu si prosím zobrazte projekt **6 – Razítka – výsledný** z připraveného studia pro učitele.

6.2.7 Testování a ladění

Cíl výuky

- Žák nalezne chyby v programu,
- opraví nalezené chyby v programu,
- testuje a ladí své programy,
- orientuje se v převzatém zdrojovém kódu.

Aktivita

- Taneční párty – chyby
- Akvárium – chyby
- Fotbal – chyby

Taneční párty - chyby

Zadání pro žáky:

Otevřete si projekt **Taneční párty – chyby** z připraveného studia.

Úloha obsahuje čtyři postavy.

Programátor v projektu udělal chyby nebo nesprávně použil některé příkazy.

Chyby nalezněte a opravte.



Obrázek 6.27: Taneční párty – chyby

Poznámky pro učitele:

Úloha obsahuje čtyři postavy.

Chlapec Boy4 se pohybuje pomocí šipek na klávesnici. U této postavy by měli žáci odhalit chybu, která byla způsobena kopírováním jednotlivých událostí a následně byla opomenuta změna v příkazu *zamiř směrem*.

Další chyba je u postavy Bear1, která by měla měnit své kostýmy podobně jako postava Cassy Dance. Zde by žáci měli přijít na chybné použití příkazu „když“ na místo příkazu „opakuj“.

Na ukázkou pro žáky nebo pro představu rozsahu projektu si prosím zobrazte projekt **7 – Taneční párty – chyby – výsledný** z připraveného studia pro učitele.

Akvárium - chyby

Zadání pro žáky:

Otevřete si projekt **Akvárium – chyby** z připraveného studia.

Tento projekt obsahuje pět postav – vodních živočichů a pozadí s mořským dnem. Při spuštění projektu můžete na první pohled zjistit, že čtyři ryby nefungují správně.

Program obsahuje chyby a nefunguje tak, jak si autor představoval. Projekt překontrolujte a opravte případné nedostatky.



Obrázek 6.28: Akvárium – chyby

Poznámky pro učitele:

U postavy Fish2 jsou chybně příkazy mimo příkaz cyklu. Tady ověříme, jestli žáci chápou, že příkazy se opakují jen tehdy, pokud jsou uvnitř těla cyklu. Také mohou zaměnit cyklus „opakuj dokola“ za cyklus s přesným počtem opakování.

U postavy Fish3 chybí příkaz *když narazíš na okraj, odraž se*. To samozřejmě není chyba, ale postava nebude dělat to, co bychom očekávali.

U postavy Shark chybí událost, která by připravený scénář spouštěla.

Postava Starfish má scénář, kde se překrývají příkazy a nenavazují na sebe (nejsou do sebe zapasovány), proto její scénář není funkční. Touto chybou jsem chtěl

upozornit na nutnost dbát na čistotu zdrojového kódu, aby se scénáře nepřekrývali a příkazy na sebe správně navazovaly.

Na ukázkou pro žáky nebo pro představu rozsahu projektu si prosím zobrazte projekt **7 – Akvárium – chyby – výsledný** z připraveného studia pro učitele.

Fotbal – chyby

Zadání pro žáky:

Otevřete si projekt **Fotbal – chyby** z připraveného studia.

Tento projekt obsahuje postavu dívky, míč a pozadí s fotbalovou brankou.

Dívka by měla přiběhnout k míči, poté uživatel stiskne klávesu mezerník a v ten okamžik by měl míč letět do branky. Následně by měla dívka jásat.

Program ovšem tak nefunguje.

Vaším úkolem je chybu najít a opravit.



Obrázek 6.29: Fotbal – chyby

Poznámky pro učitele:

Dívka by měla přiběhnout k míči, poté uživatel stiskne klávesu mezerník a v ten okamžik by měl míč letět do branky. Následně by měla dívka jásat.

Chyby jsou v této úloze dvě a týkají se posílání zpráv mezi postavami.

Obě chyby jsou u postavy Soccer Ball. V příkazu *po obdržení zprávy* je chybně vybrána zpráva *gol* (má být vybrána zpráva *kopni*) a na konci toho samého scénáře chybí odeslání zprávy *gol*.



Obrázek 6.30: Fotbal – chyby – obě chyby jsou ve scénáři postavy „Soccer Ball“

Pokud by mnou vytvořené úlohy vyučujícímu nestačily, pak je možné využít úlohy připravené týmem ScratchEdTeam, které jsou dostupné na následujících odkazech:

<http://scratch.mit.edu/projects/2042697/>,

<http://scratch.mit.edu/projects/2042703/>,

<http://scratch.mit.edu/projects/2042706/>,

<http://scratch.mit.edu/projects/2042712/>

<http://scratch.mit.edu/projects/2042724/>,

nebo novější verze připravených projektů „Debug-It“

<http://scratch.mit.edu/projects/10437040/>,

<http://scratch.mit.edu/projects/10437249/>

<http://scratch.mit.edu/projects/10437366/>,

<http://scratch.mit.edu/projects/10437439/>

<http://scratch.mit.edu/projects/10437476/>

6.2.8 Klonování

Cíl výuky

- Žák vytvoří postavu, kterou později použije pro klonování,
- naprogramuje scénář, který bude provádět postava po startu jako klon,
- ve scénáři opakovaně klonuje připravenou postavu
- dokáže zrušit klon.

Aktivita

- Myši
- Arkanoid – příprava hry
- Chytání švestek – příprava hry

Myši

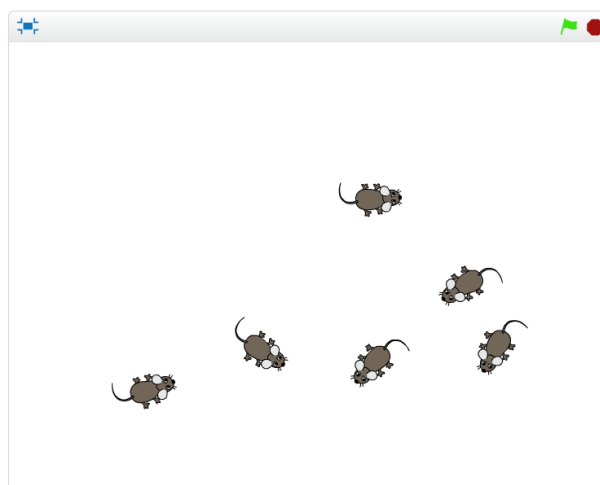
Zadání pro žáky:

Otevřete si editor Scratch a v něm nový projekt.

Vaším úkolem je vytvořit program, kde se budou množit myši.

Myši se budou náhodně objevovat a budou běhat po scéně.

Nejprve vložte do scény postavu (myš). Tuto postavu budete klonovat. Vznikající klony se budou chaoticky pohybovat po scéně.



Obrázek 6.31: Myši

Poznámky pro učitele:

Úloha si klade za cíl na jednoduchém příkladu seznámit žáky s klonováním. Tento program by pak mohli dále rozšiřovat. Podstatné je, aby si žáci uvědomili, že klon

postavy má svůj scénář, který se začne provádět ihned po startu klonu. Také musejí umět použít příkaz pro vytvoření klonu (příkaz *klonuj*). Můžeme udělat prakticky neomezené množství klonů a lze je také zrušit.

Na ukázkou pro žáky nebo pro představu rozsahu projektu si prosím zobrazte projekt **8 – Myši – výsledný** z připraveného studia pro učitele.

Arkanoid – příprava hry

Zadání pro žáky:

Otevřete si editor Scratch a v něm nový projekt.

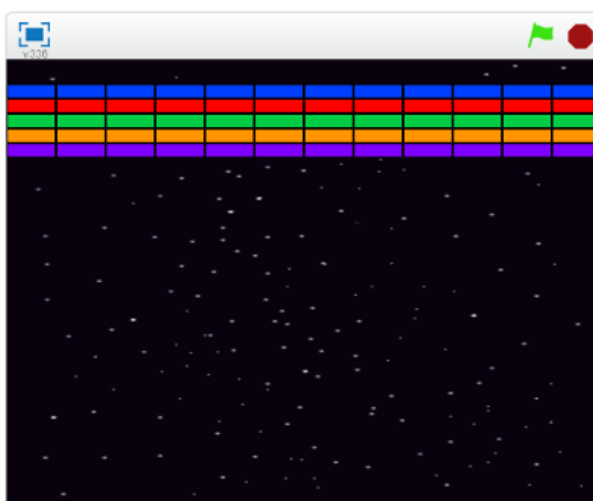
Vaším úkolem je vytvořit základ pro hru Arkanoid.

Do scény nejprve vložte vhodné pozadí.

Připravte si jednu cihličku (postavu) a tu posléze naklonujte na vhodné pozice ve scéně.

Udělejte nejméně pět řad a každou řadu jinou barvou podobně, jak ukazuje obrázek.

V následujících tématech budeme projekt dále rozvíjet.



Obrázek 6.32: Arkanoid

Poznámky pro učitele:

Úloha je přípravou pro hru Arkanoid. Žáci by měli být schopni naklonovat jednu postavu (cihličku) na patřičná místa ve scéně.

Žáky bychom měli vést k tomu, aby pro klonování využili dva cykly. První se bude starat o změnu kostýmu a posun v ose y, druhý cyklus (vnořený) o samotné klonování a posun v ose x.

Na ukázkou pro žáky nebo pro představu rozsahu projektu si prosím zobrazte projekt **8 – Arkanoid – výsledný** z připraveného studia pro učitele.

Chytání švestek – příprava hry

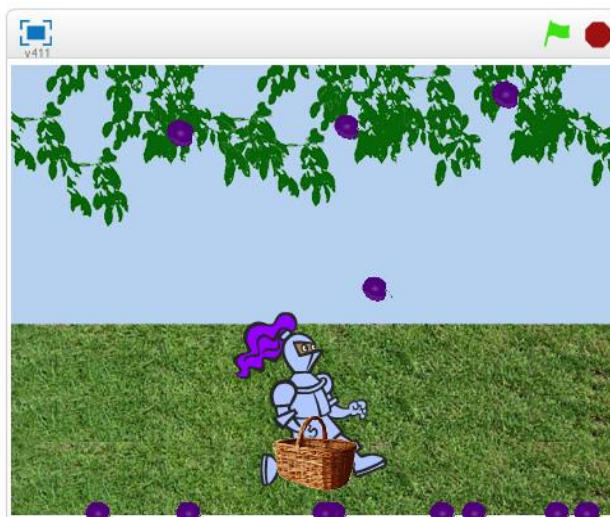
Zadání pro žáky:

Otevřete si projekt **Chytání švestek** z připraveného studia.

Projekt obsahuje pozadí a tři postavy – rytíře, košík a švestku.

Vaším úkolem je upravit tento připravený projekt tak, aby se v horní části scény náhodně objevovaly švestky a po nějaké době padaly dolů na zem. Využijte postavu švestka, pro kterou vytvoříte vhodný scénář.

Postavy v dolní části scény zatím nijak neupravujte, později je naprogramujete k chytání švestek do košíku.



Obrázek 6.33: Chytání švestek

Poznámky pro učitele:

Opět se jedná o přípravu jednoduché hry, která zároveň poslouží k procvičení klonování.

Žáci mohou využít připravené scény a jen naprogramovat klonování švestek.

Na ukázkou pro žáky nebo pro představu rozsahu projektu si prosím zobrazte projekt **8 – Chytání švestek – výsledný** z připraveného studia pro učitele.



Obrázek 6.34: Klonování postavy „švestka“

6.2.9 Podmíněný příkaz a operátory

Cíl výuky

- Žák chápe větvení programu,
- vloží do scénáře podmíněný příkaz – vhodnou variantu příkazu „když“,
- u příkazu „když“ nastaví podmínku s pomocí vnímání a operátorů,
- ve svých projektech využívá příkaz „když“ a jeho variantu „když - jinak“.

Aktivita

- Chytání švestek
- Arkanoid

Chytání švestek

Zadání pro žáky:

Otevřete si rozpracovaný projekt z předchozího tématu.

Padající švestky bychom nyní potřebovali chytat do košíku.

Pokud se padající švestka dotkne košíku, košík ji do sebe pojme a švestka zmizí.

Využijte podmíněný příkaz „když“ a příkazy vnímání k dopracování tohoto efektu.

Později k projektu přidáme počítání chycených švestek.



Obrázek 6.35: Chytání švestek

Poznámky pro učitele:

Žáci by v této úloze měli využít příkaz *když* a příkaz vnímání k ověření, jestli se švestka dotýká košíku. Toto ověření bude prováděno ve scénáři klonů. Na základě vyhodnocení této podmínky dojde ke zrušení klonu (obrázek 6.36). Žáci mohou dělat

chyby v tom, že neumístí podmíněný příkaz do těla cyklu, ale například před cyklus nebo za něj. Další chyba by mohla být kontrola dotyku se švestkou u jiné postavy. Potom by byl problém klon švestky zrušit. Později s pomocí proměnných doplníme počítání bodů.

Na ukázkou pro žáky nebo pro představu rozsahu projektu si prosím zobrazte projekt **9 – Chytání švestek – výsledný** z připraveného studia pro učitele.



Obrázek 6.36: Podmíněný příkaz ve scénáři klonu postavy „švestka“

Arkanoid

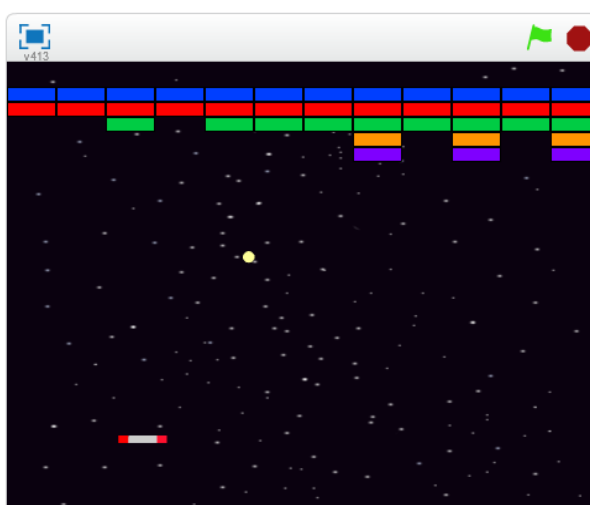
Zadání pro žáky:

Otevřete si projekt Arkanoid, který máte rozpracovaný z předchozího tématu.

Máme připravené pouze naklonované cihličky v horní části scény.

Vaším úkolem je přidat do scény kuličku spolu s tyčkou a naprogramovat jejich scénáře.

Kulička se bude odrážet od tyčky, okrajů scény a také od barevných cihliček, které bude zároveň bořit (cihličky budou mizet).



Obrázek 6.37: Hra Arkanoid

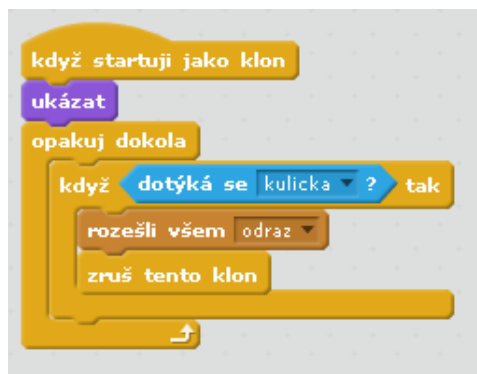
Poznámky pro učitele:

Žáci by v této úloze měli přidat dvě postavy – kuličku a tyčku. Pro tyčku budou muset naprogramovat pohyb doleva a doprava. Kulička by se měla pohybovat neustále a odrážet se od tyčky, od cihliček a od okrajů scény.

Žáci budou muset využít příkaz *když* a příkaz *dotýká se* pro odraz kuličky od tyčky. Podobně cihlička musí odrazit kuličku a zmizet (klon bude zrušen).

Nesprávným řešením může být například chybné testování dotyku kuličky s cihličkou. V případě testování u postavy kuličky by následně nedocházelo k odrazům kuličky od cihliček. Chyb ve scénářích tohoto programu může být celá řada, proto na obrázcích 6.38 a 6.39 ukazují nejdůležitější části scénářů.

Na ukázkou pro žáky nebo pro představu rozsahu projektu si prosím zobrazte projekt **9 – Arkanoid – výsledný** z připraveného studia pro učitele.



Obrázek 6.38: Hra Arkanoid – podmíněný příkaz ve scénáři klonu cihličky



Obrázek 6.39: Hra Arkanoid – podmíněný příkaz ve scénáři kuličky

6.2.10 Proměnné

Cíl výuky

- Žák vytvoří proměnné a globální proměnné,
- přiřadí hodnotu do proměnné,
- mění hodnotu proměnné,
- vytvoří seznam a pracuje s ním,
- aktivně používá proměnné ve svých projektech.

Aktivity

- Chytání švestek – skóre a životy
- Arkanoid – skóre a životy
- Kalkulačka

Chytání švestek – skóre a životy

Zadání pro žáky:

Otevřete si projekt **Chytání švestek**, který máte rozpracovaný z předchozího tématu.

Nyní rozšíříme tento projekt o počítání chycených švestek do košíku a také omezíme životy (kolik švestek nám může spadnout na zem). V případě že hráč vyčerpá všechny životy, hra bude ukončena.

Skóre i životy se pokuste naprogramovat pomocí proměnných.



Obrázek 6.40: Chytání švestek

Poznámky pro učitele:

Žáky seznámíme s vytvořením a používáním proměnných.

Dále ukážeme, jaký je rozdíl mezi „obyčejnou“ proměnnou a proměnnou uloženou na serveru (cloud proměnná). Právě pomocí této proměnné lze uchovávat rekord hry. Pomocí proměnných by žáci měli doplnit do hry skóre a také počet životů. V případě, že se švestka dotkne košíku, připočítá se jeden bod do proměnné a zruší se klon švestky. Když švestka spadne na zem, je potřeba odečíst jeden život. Na ukázkou pro žáky nebo pro představu rozsahu projektu si prosím zobrazte projekt **10 – Chytání švestek – výsledný** z připraveného studia pro učitele.



Obrázek 6.41: Chytání švestek – část scénáře klonu švestky s proměnnými

Arkanoid – skóre a životy

Zadání pro žáky:

Otevřete si projekt Arkanoid, který máte rozpracovaný z předchozího tématu.

Tuto hru také vylepšete s pomocí proměnných o skóre (počítání zasažených cihliček) a omezte počet životů.

Pokuste se hru rozšířit například o řízení rychlosti kuličky. Kulička by mohla v závislosti na skóre zrychlovat.

V následujícím tématu naprogramujete postup do dalšího levelu.



Obrázek 6.42: Arkanoid

Poznámky pro učitele:

Hra Arkanoid může posloužit na procvičení používání proměnných.

V proměnných mohou žáci uchovávat například počet zbývajících životů, skóre, rekord hry a také aktuální rychlost kuličky.

Pokud žáci budou chtít řídit rychlost kuličky, použijí proto také proměnnou. Mohou například při každém odrazu kontrolovat skóre, a když dosáhne určité hodnoty, zvýší hodnotu v proměnné, která uchovává rychlost kuličky.

Na ukázkou pro žáky nebo pro představu rozsahu projektu si prosím zobrazte projekt **10 – Arkanoid – výsledný** z připraveného studia pro učitele.

Kalkulačka

Zadání pro žáky:

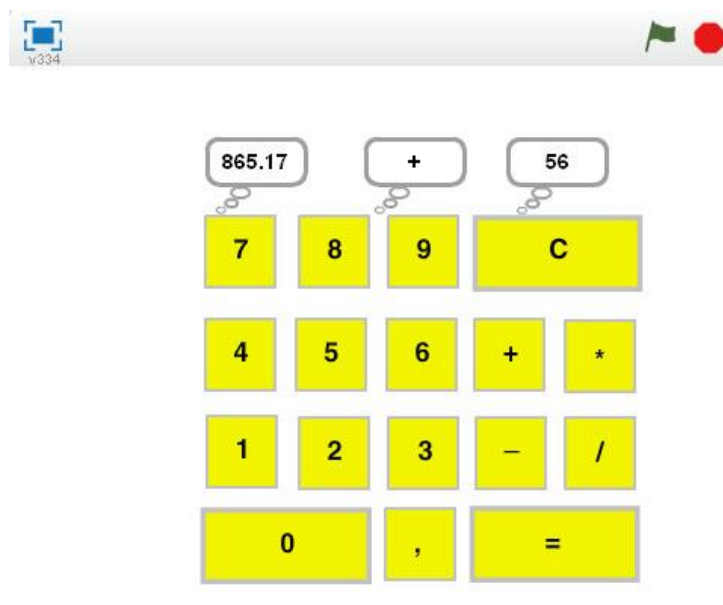
Otevřete si editor Scratch a v něm nový projekt.

S pomocí proměnných vytvořte program, který bude fungovat jako jednoduchá kalkulačka.

Kalkulačka bude umět sčítat, odečítat, násobit a dělit.

Vytvořte jednotlivá tlačítka kalkulačky, na která bude možné klikat myší a tím zadávat hodnoty, případně operace s čísly.

Pro uchování zadávaných čísel použijte seznam.



Obrázek 6.43: Kalkulačka

Poznámky pro učitele:

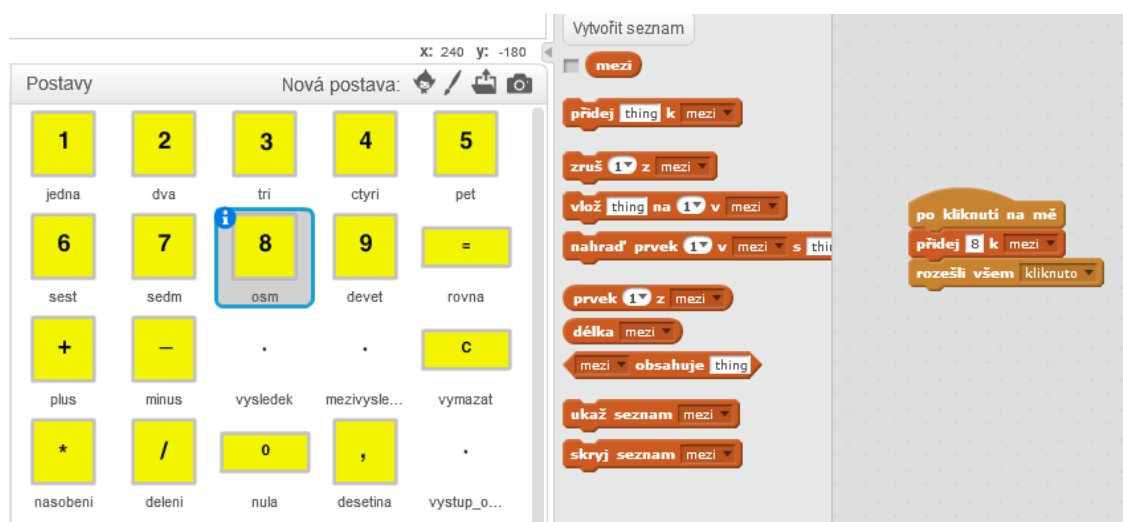
V této úloze si žáci procvičí jak proměnné, tak posílání zpráv.

Tato úloha by mohla také posloužit, jako seznámení s proměnnou „seznam“ (jedná se o pole).

Úlohu lze řešit jednodušším způsobem, když si žáci nechají čísla zadávat od uživatele pomocí příkazu *ptej se*.

Pokud žáci budou chtít zadávat čísla klikáním na tlačítka kalkulačky, nevyhnou se použití právě seznamu.

Na ukázkou pro žáky nebo pro představu rozsahu projektu si prosím zobrazte projekt **10 – Kalkulačka – výsledný** z připraveného studia pro učitele.



Obrázek 6.44: Seznam u kalkulačky

6.2.11 Bloky (Procedury)

Cíl výuky

- Žák definuje procedury,
- používá procedury,
- vyhodnotí situaci v programu a určí, zda je vhodné použít proceduru nebo nikoliv.

Aktivita

- Arkanoid – další levely
- Závod

Arkanoid – další levely

Zadání pro žáky:

Otevřete si projekt **Arkanoid**, který máte rozpracovaný z předchozího tématu.

Najděte ve vytvořené hře část příkazů, které se vyskytují v programu opakovaně, nebo část která provádí určitou činnost, která by stála za to pojmenovat a tím zpřehlednit zdrojový kód. Z nalezené části vytvořte proceduru a tu potom budete volat v programu.

Ve hře může být také implementována procedura, která bude zajišťovat postup do vyšší úrovně.



Obrázek 6.45: Arkanoid

Poznámky pro učitele:

Necháme žáky projít zdrojový kód hry a nalézt část příkazů, ze kterých by mohli vytvořit proceduru. Aby mělo smysl proceduru vytvářet, mělo by se jednat o část programu, která se někde opakuje. Procedury také přispějí k přehlednosti celého programu.

Proceduru lze také velmi dobře použít na ověření dosažených bodů a zvýšení úrovně hry (levelu).

Na ukázkou pro žáky nebo pro představu rozsahu projektu si prosím zobrazte projekt **11 – Arkanoid – výsledný** z připraveného studia pro učitele. Podívejte se prosím na scénáře scény, kde jsou provedeny největší změny.



Obrázek 6.46: Arkanoid – kontrola a zvýšení levelu

Závod

Zadání pro žáky:

Otevřete si editor Scratch a v něm nový projekt.

Vytvořte závodní hru, ve které využijete procedury.

Tratí závodu může být např. jednoduchý ovál.

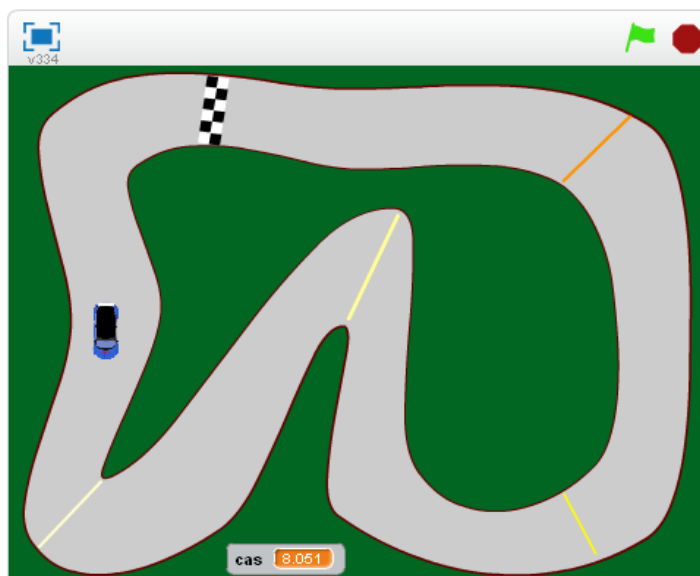
Uživateli umožněte ovládání závodního auta pomocí klávesnice, auto může i zrychlovat a zpomalovat.

Dokážete změřit čas, po který bude závodník na trati?

Lze uchovávat rekordní čas?

Hru následně můžete vylepšovat například o kontrolní průjezdní body, nebo zajistit zpomalení auta na ploše mimo silnici.

Příkladem použití procedury můžete být právě zajištění této kontroly – průjezdních bodů, auta mimo silnici, nebo dojezd do cíle.

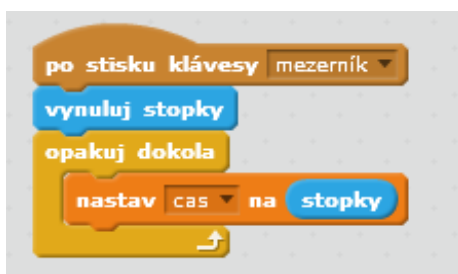


Obrázek 6.47: Závod

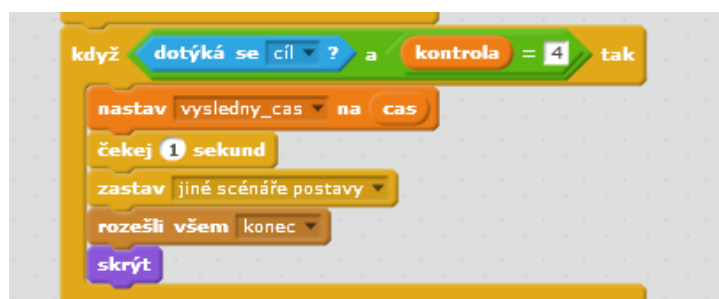
Poznámky pro učitele:

V této úloze žáci použijí všechny doposud získané znalosti – procedury, proměnné, podmíněný příkaz, posílání zpráv a další. Mohou také použít stopky ke změření doby, po kterou bude auto projíždět tratí (obrázek 6.47).

Na ukázkou pro žáky nebo pro představu rozsahu projektu si prosím zobrazte projekt **11 – Závod – výsledný** z připraveného studia pro učitele.



Obrázek 6.48: Závod – použití stopek pro měření času



Obrázek 6.49: Závod – část scénáře auta – kontrola dojetí do cíle

6.2.12 Závěrečný projekt

Cíl výuky

- Žák pracuje na samostatném projektu,
- spolupracuje s žáky v týmu,
- vymýšlí originální obsah projektu,
- vytvoří osnovu projektu,
- vytvoří grafiku do projektu,
- naprogramuje scénáře dle připravené osnovy.

Zadání pro žáky:

Vaším úkolem je zpracovat rozsáhlejší projekt, například vlastní hru.

Nejprve si rozmyslete, na jaké téma budete projekt zpracovávat a následně si zpracujte scénář projektu. Až budete mít vše dobře rozmyšlené, můžete se pustit do vytváření grafiky a programování.

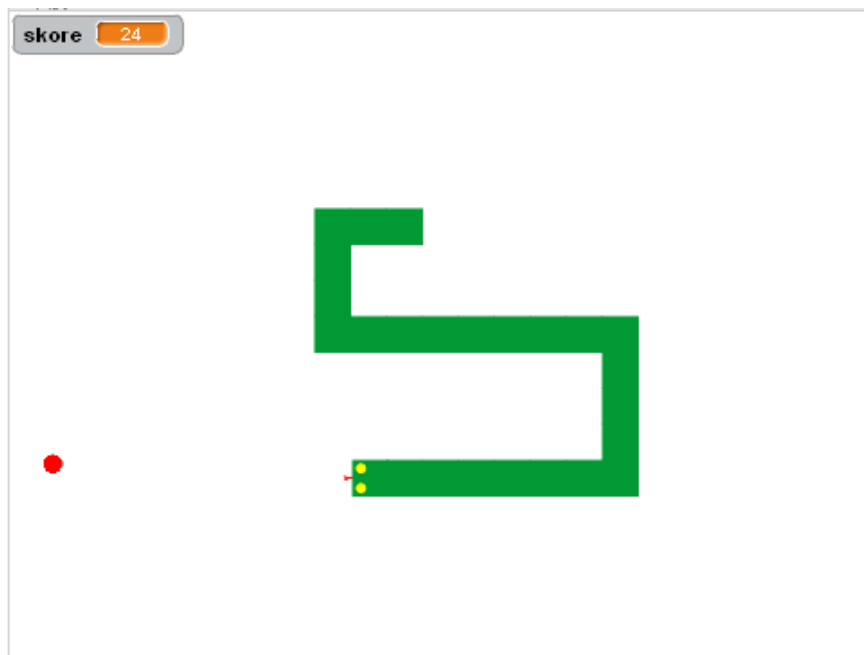
Poznámky pro učitele:

Nejprve rozdělíme žáky do týmů. Necháme každému týmu prostor na brainstorming nápadů na téma projektu. Každá skupina by měla mít nejprve jasnou představu o tom, co chtějí dělat. Potom začnou plánovat, udělají osnovu projektu. Dále zpracují grafiku a budou programovat.

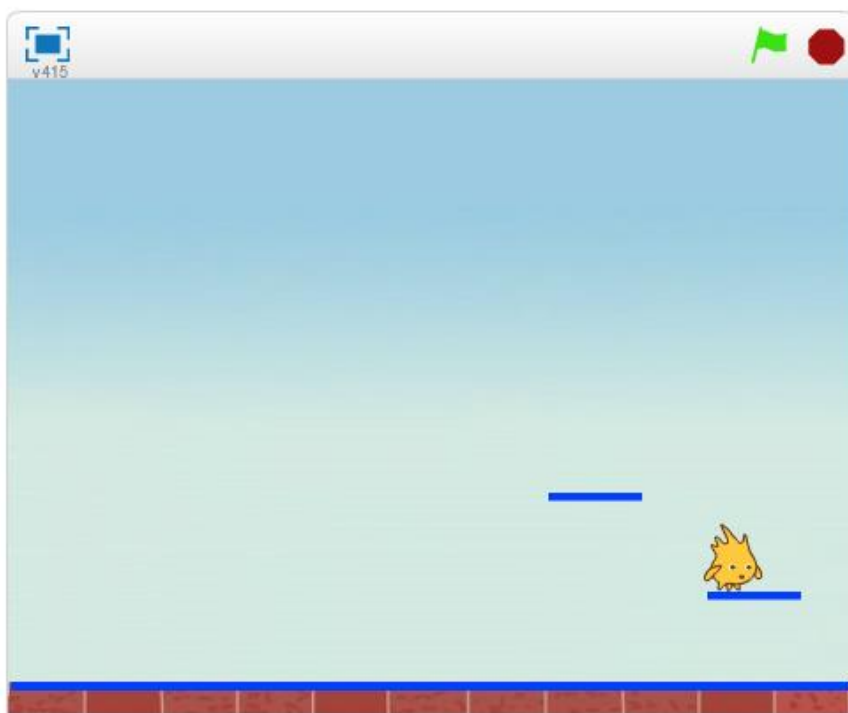
V týmu by měli vzniknout role, také si členové týmu musejí mezi sebe rozdělit úkoly. Měli bychom na žáky apelovat, aby pracovali postupně po malých částech programu a všechno testovali a ladili.

Se svým projektem pak mohou žáci předstoupit před třídu a ukázat jej ostatním. Mohou svůj projekt také sdílet s ostatními přímo na portálu Scratch.

Pokud bude učitel chtít zobrazit žákům vzorové projekty, připravil jsem pro tuto příležitost dvě hry, které jsou připravené ve studiu pro učitele:



Obrázek 6.50: projekt 12 – had



Obrázek 6.51: projekt 12 – Jumping to star

7. Ověření postupu výuky v praxi

Ověření výsledků své diplomové práce jsem provedl na dvou úrovních. První a zároveň dlouhodobější ověření jsem provedl v rámci zájmového kroužku, kterého se zúčastňují žáci prvních ročníků střední školy z různých oborů, maturitních i učebních. Druhé ověření jsem provedl v rámci projektových dnů, které pořádá naše škola pro deváté ročníky základních škol.

7.1 Ověření v rámci zájmového kroužku

Zájmového kroužku se zúčastňovalo obvykle šest až deset žáků z prvních ročníků střední školy z různých oborů a to pouze chlapci. Výuka v rámci zájmového kroužku probíhala dvacet vyučovacími hodinami. Ve výuce jsme postupovali podle materiálů, které jsem připravil. Jako metodu ověření výuky jsem použil zúčastněné pozorování, analýzu žákovských prací a rozhovor s žáky po absolvování výuky.

Seznámení s prostředím Scratch

V úvodu jsem žáky seznámil s webovými stránkami projektu Scratch, se samotným prostředím a všichni se zaregistrovali. V první úloze (s názvem První program) žáci neměli žádný problém s vložením další příkazu, či sestavováním scénářů a okamžitě se začali projevovat jejich tvůrčí schopnosti. Scratch tady svou jednoduchostí a intuitivností sám vybízí k samostatnému tvoření. Žáci vytvořili s postavou celou řadu zajímavých scén, za všechny budu jmenovat jen některé. Například postava přišla, pozdravila, zeptala se na to „jak se máte“, po obdržení odpovědi se otočila a odešla. V dalším projektu byli dvě postavy, které na stisk klávesy mezerník reagovaly posunem. Jeden žák dokonce udělal scénu, ve které se postava po stisku klávesy posunula ke kurzoru myši.

Posloupnost příkazů a cyklus

Na druhém setkání jsem v úvodu chtěl, aby žáci jmenovali situace z běžného života, které mají svůj řád a danou posloupnost kroků. Žákům to dělalo problémy a nakonec jmenovali pouze návod na uvedení do provozu elektrického spotřebiče a recept na vaření.

Velmi přínosné pro pochopení problematiky bylo, když jeden ze žáků dával povely druhému a ten je vykonával.

Pohyb postav v programu Scratch, který by symbolizoval tanec, nedělal žákům žádný problém, dokonce se objevila řada různých řešení, například interaktivních pomocí různých kláves.

Zajímavé bylo, na rozdíl od výuky se žáky ze základních škol, že nám dalo více práce najít příkaz *opakuj*. U úlohy Mlsný kocour jsme objevili chybu, kterou jsem později opravil. Žáci využívali příkaz *opakuj* a pochopili smysl cyklů v programování.

Postavy a pozadí

Ke třetímu cvičení jsem jen ukázal, kde změnit či přidat pozadí scény, případně kostýmy postavám. Dále již byli žáci schopni samostatně měnit kostýmy postavám. V úlohách se procvičoval cyklus, změny kostýmů a pozadí. Během cvičení jsem žákům ukázal jak kopírovat bloky příkazů jiným postavám, což velmi uvítali. Někteří používali pro plynulý pohyb postav klouzání, později sami nahradili tento pohyb použitím cyklu.

V průběhu cvičení jsem došel k tomu, že čtyři úlohy, které jsem připravil pro toto téma, je na jednu hodinu příliš. Úlohy procvičují podobné příkazy a není potřeba využívat všechny. Učitelé si mohou zvolit, které úlohy využijí.

Velmi se mi osvědčila úloha, která navazuje na předchozí úlohu **Mlsný kocour 2**. Žáci si v této úloze opět procvičili použití cyklu. Zmizení pečinky a barevný efekt u postavy kocoura žáci řešili vhodným časováním, bez použití jakýchkoliv podmínek a podmíněných příkazů.

Více postav a postupný vývoj

V rámci čtvrtého pracovního listu měli žáci naprogramovat scénu **Taneční párty**, jejíž námět vychází z materiálu Scratch curriculum guide od Karen Brennan [15].

Žáci pochopili, v jakém případě je vhodné použít postavu, kdy kostým a byli schopni říci, jaký je mezi nimi rozdíl. Bylo velice dobré, že se tyto pojmy vyjasnili pro celou skupinu a nikdo s tím dále neměl potíže. Nebylo potřeba, abych cokoliv ukazoval, ihned se všichni pustili do práce.

Největším lákadlem pro žáky bylo vložit a přehrávat ve scéně oblíbené písně. Někteří se vyfotili a použili svou fotku jako jednu z postav. U této úlohy byli velmi kreativní, úloha je velice bavila a naplnila cíle výuky.

Paralelizace a události

Páté cvičení jsme začali opakováním cyklů a vyzkoušeli si, jak funguje nekonečný cyklus. Také jsem na začátku ukázal na dvou libovolných postavách fungování zelené

vlažky. Dále byli žáci schopni vytvořit různé animace, rozhodně jsem je neomezoval ani nevyžadoval **animaci akvária**. Přišli na to, že postavy mohou nezávisle na sobě vykonávat různé činnosti. Jediné co jsem vyžadoval, aby si zkusili, jestli i jedna postava může dělat více činností najednou. Úloha splnila své cíle a od žáků začaly zaznívat první nápady na hry, které by chtěli vytvořit.

Další úloha na procvičení **Kouzelná hůlka** byla z mého pohledu prospěšná tím, že žáci museli přijít na to, že hůlka je postava a jakým způsobem se bude pohybovat spolu s kurzorem myši. Další náležitosti scénářů jim nečinili žádné větší potíže a úlohu zvládli naprogramovat.

Události a zprávy

Výuku jsem zahájil tím, že jsem ukázal na jedné postavě, jak lze udělat, aby při kliknutí či stisknutí klávesy postava přehrála zvuk, nebo vykonala pohyb. Potom jsem nechal žáky, aby si zvolili z prvních dvou úloh jednu a tu naprogramovali – **Hudební nástroj**, nebo **Bludiště**. Žáci zvolené úlohy pojali po svém a dokázali samostatně vytvořit zajímavé projekty. Pozorováním žáků jsem zjistil, že úloha na vytvoření hudebního nástroje byla o něco atraktivnější. Někteří žáci stihli obě úlohy.

Na úloze **Vypínač** jsem ukázal posílání zpráv. Nechal jsem žáky přicházet na to, jak kliknutím na postavu mohou změnit například pozadí. Někteří na řešení pomocí posílání zpráv přišli velmi brzy, jiní potřebovali pomoc. Při výuce jsem si ověřil, že učitel velmi snadno improvizuje a do každé úlohy si může vymyslet nějaký úkol navíc, kterým dokáže udržet pozornost i nadanějších žáků.

Poslední úlohu **Razítka** stihli jen někteří žáci, ale i přesto byla přínosná k procvičení používání událostí i zpráv.

Testování a ladění

Sedmé téma je zaměřené na orientaci ve zdrojovém kódu, hledání chyb v projektech a následnou úpravu. Tady je důležité být trpělivý a nechat žáky co nejdéle programy zkoumat a hledat chyby.

V první úloze **Taneční párty – chyby**, podobně jako v ostatních dvou, žákům chvíli trvalo, než se ve zdrojovém kódu zorientovali, ale pak chyby odhalili celkem bez větších problémů. Velmi jim pomohlo, když jsem jim řekl, kolik chyb mají vlastně hledat. Možná pro ně byla trochu matoucí ta část scénáře, kterou ještě neznají - použití

podmíněných příkazů. V této části však žádné chyby nebyly a úloha tím byla atraktivnější.

Ve druhé úloze **Akvárium – chyby**, která je obdobou úlohy akvárium, jsou některé chyby hůře odhalitelné a také je jich tam více. Někteří žáci potřebovali malou nápovědu, aby přišli na všechny chyby.

Vzhledem k tomu, že na kroužek docházejí jen chlapci, třetí úloha **Fotbal – chyby**, především díky fotbalové tematice, byla pro ně velice atraktivní. Opět platilo, že někteří odhalili chyby ihned a někteří potřebovali nápovědu, že se jedná o chybu v posílání zpráv.

Když bude některý z žáků hotov dříve, může ho učitel nasměrovat na úlohy od autorů ScratchEdTeam.

Klonování

Další téma bylo zaměřeno na klonování postav. Se žáky jsem začal první úlohou s názvem **Myši**. U této úlohy se ukázalo, že se jedná o složitější téma. Žáci jsou samozřejmě schopni najít příkazy pro klonování, ale málokterý z nich přišel na to, jak správně klonování používat. Proto jsme první úlohu udělali společně.

Druhá úloha s názvem **Arkanoid**, byla pro žáky také náročnější, ale velmi dobře si procvičili klonování i cykly. Někteří žáci zvládli úlohu naprogramovat, ale někteří potřebovali pomoc nebo malou nápovědu. Proto si myslím, že je dobře zařadit na procvičení ještě další úlohu.

Využil jsem také třetí úlohu z pracovního listu s názvem **Chytání švestek**. S touto úlohou si žáci poradili o poznání lépe.

Myslím, že žáci na základě připravených úloh klonování pochopili a dokážou ho používat.

Podmíněný příkaz a operátory

V rámci devátého tématu zaměřeného na podmíněné příkazy se mi potvrdilo, že se jedná o obtížnější téma. Bylo zapotřebí použití podmíněného příkazu žákům nejprve ukázat spolu s příkazy vnímání. Doporučuji ukázat podmíněný příkaz a vnímání na nějaké postavě a potom využít úlohy na procvičení. Popřípadě použít první úlohu jako ukázkou a druhou nechat žákům dělat samostatně. Bylo zapotřebí chodit mezi žáky a občas někomu poradit.

Přibližně polovina žáků zvládla projekty naprogramovat. Ostatní žáci si s oběma projekty poradili malou pomocí a nápovědou.

V rámci procvičování jsem přišel na chybu v mém projektu týkajícího se padajících švestek. Chybu jsem následně opravil u všech třech sdílených projektů ve studiu.

Proměnné

Téma proměnných nedělalo žákům takové potíže jako témata předchozích kapitol. První úlohu **Chytání švestek** jsem využil, abych žákům ukázal, jak s proměnnými pracovat.

Druhou hru **Arkanoid** už žáci vylepšovali sami, podle svých představ. Opět bylo nutné některým žáků maličko poradit.

Rychlejší žáci si vyzkoušeli udělat i jednoduchou **kalkulačku**. I přesto, že to není nejatraktivnější úloha, na procvičení proměnných byla velice dobrá. Jeden z chlapců pro naprogramování kalkulačky vytvořil vlastní seznam a zkoušel s ním pracovat.

Procedury

V úvodu jedenáctého setkání jsem zmínil, že blokem (procedurou) může být ta část scénáře, která se bude vícekrát opakovat, můžeme mu předávat proměnné a má velký význam ve zpřehlednění samotného scénáře.

Na připravené hře **Arkanoid**, jsem ukázal možné použití Bloků.

Úloha Závodní hra chlapce velmi bavila, jednalo se však o obsáhlejší projekt, který potřeboval více času. V rámci kroužku jsme mu mohli věnovat dvě vyučovací hodiny.

Závěrečný projekt

Poslední téma s názvem Závěrečný projekt jsem s žáky v rámci kroužku pouze zahájil. Žáky jsem rozdělil do dvou týmů a nechal je rozmyslet téma projektu. Na projektu budou týmy dále samostatně pracovat.

Závěry z ověření výuky v rámci zájmového kroužku

V rámci zájmového kroužku jsem ověřil, že navržená metodika lze použít pro výuku základů programování. Tvoření v prostředí Scratch žáky bavilo a podle ohlasů byla tato výuka pro ně příjemným zpestřením. Při svém ověření jsem odzkoušel, že každé téma lze stihnout v rámci jedné vyučovací hodiny, samozřejmě kromě

závěrečného projektu. Pokud by učitel chtěl více procvičovat a případně nechat žákům větší prostor na vlastní tvoření, lze použít jednotlivá témata i pro dvouhodinovky.

Je zřejmé, že pokud se v rámci informatiky najde prostor pro zařazení výuky programování, bude to pravděpodobně v rámci jednoho pololetí a tudíž reálně šestnáct až sedmáct vyučovacích hodin, což považuji za minimální, ale dostatečnou časovou dotaci.

7.2 Výuka Scratch v rámci projektových dnů

Naše škola uspořádala pro žáky devátých ročníků základních škol z Českých Budějovic, Zlivy a Hluboké nad Vltavou několik projektových dnů, kterých se zúčastnili čtyři základní školy a celkem 149 žáků. Přijeli k nám na čtyři vyučovací hodiny, v rámci každé z nich absolvovali jiné aktivity.

Jednu z aktivit jsem měl možnost vést já s cílem **představit žákům prostředí Scratch**. Předpokládal jsem, že na těchto školách se v rámci informatiky nevyučuje programování a žáci neznají Scratch. Tento předpoklad se naprosto potvrdil.

Dalším mým cílem bylo seznámit účastníky s **cykly** (variantami příkazu *opakuj*) a s podmíněným příkazem. Vzhledem k tomu, že jsem měl na výuku pouze 45 min, tak nebylo možné dopředu odhadnout, jestli jsou mé cíle reálné a dosažitelné.

V úvodu výukové hodiny jsem účastníkům představil webové stránky Scratch a navedl je na spuštění několika projektů. V projektech bylo možné navázat ukázkou, jak si mohou zobrazit zdrojový kód projektu. Pak jsem v rychlosti představil prostředí a začali jsme s tvořením jednoduchého scénáře. Vždy jsem využil postavu, se kterou jsem posunoval (v podstatě to bylo cvičení **Tanec postavy**). Snažil jsem se žáky navést na to, že mohou kroky opakovat. U většiny skupin se mi povedlo docílit toho, že na použití příkazu *opakuj* přišli samostatně. Jako další jsem ukázal příkaz *opakuj dokola*.

Dále jsem žákům navrhnul, že společně vytvoříme libovolnou animaci, jako například **akvárium**, animaci kopnutí míče do brány, nebo tanec několika postav – **Taneční párty** (všechny programy jsou dostupné v mém studiu <http://scratch.mit.edu/studios/202382/>).

Žáci dostali prostor být kreativní, slovně jsem je k tomu vybízel. V žádném případě jsem nevyžadoval, aby tvořili to samé, co jsem já předváděl. Většinou si tvořili svou animaci, ale někteří dělali to samé co já. Úkolem bylo vytvořit animaci – použít více postav, které se měli pohybovat a některé na sebe měli reagovat. Animace mohla

být (a většinou byla) interaktivní. Žáci používali různé příkazy, ale především cykly. Scénáře kopírovali všem postavám. Řešení úkolu reakce postav mezi sebou, se podařilo navrhnout jen málokomu, ale když jsem ukázal jak na to, pak to byli schopni udělat všichni žáci (problém jsme řešili s pomocí podmíněného příkazu *když* a příkazů *vnímání*).

Závěry z ověření výuky v rámci projektových dnů

Výukou programu Scratch v rámci projektových dnů jsem si ověřil, že Scratch je velmi intuitivní, uživatelsky přívětivý program a velmi vhodný pro použití při výuce programování pro děti. Všichni žáci devátých tříd, kteří v programu Scratch pracovali poprvé, byli schopni během 45 minut naprogramovat jednoduchou animaci, někteří i více. Drtivá většina žáků se žádným způsobem nikdy programování nevěnovala. Ze všech účastníků byli pouze dva žáci, kteří přiznali, že se programování věnují, z toho jeden profesionálně a to konkrétně jazyku Java.

Později jsem se dozvěděl, že žákům se má hodina v rámci projektového dne líbila a byla pro ně zajímavá. Podobně se o mé výuce vyjádřil i pan ředitel z jedné Českobudějovické základní školy, který u mě strávil celý projektový den (4 vyučovací hodiny).

Žáci, kteří se zúčastnili projektových dnů a prošli mou hodinou, byli ze čtyř základních škol, chlapci i dívky v celkovém počtu 149 žáků. To považuji za dostatečně velký vzorek žáků z okolí, abych z toho dokázal vyvodit některé závěry. Na základě mého ověření výuky programování v prostředí Scratch se mi podařilo vyvrátit tvrzení některých kolegů ze základních škol, a to například „Žáci naší školy by toto učivo nezvládli“, nebo „Je nad možnosti žáků.“, které uvedli v mém dotazníkovém šetření. Z průzkumu a ověření výuky je zřejmé, že někteří učitelé informatiky na základních školách své žáky podceňují a ochuzují je tím o témata, která by evidentně byli schopni zvládnout. Na základě mých zjištění tedy tvrdím, že žáci základních škol jsou schopni učit se programování a mohlo by být zařazeno do školních vzdělávacích programů.

8. Závěr

Scratch je ve světě jedním z vedoucích programů ve výuce programování pro děti. Porovnáním programu Scratch s Baltíkem a Imaginem jsem zjistil, že Scratch v ničem nezaostává za těmito programy, které se u nás běžně používají a mohl by se do vedoucí pozice dostat i u nás. Podpůrných materiálů pro výuku programování ve Scratch stále přibývá. Zpracoval jsem porovnání metodik, které jsou dostupné z portálu scratched.media.mit.edu. Začínají se objevovat i české publikace. Bohužel jsem nestihl tyto nové materiály do své práce zahrnout.

Vytvořil jsem vzorový projekt, na kterém demonstruji většinu příkazů, které chceme žáky v průběhu výuky naučit. Osnovu výuky, metodické listy a žákovské listy jsem vytvořil pro podporu výuky programování na základních školách. Tyto listy doplňují dvě studia, která jsou připravena pro žáky a učitele. Ověřením těchto materiálů jsem zjistil, že žáci jsou schopni učit se základy programování podle mnou připravené výuky.

Navíc nad rámec zadání jsem provedl průzkum na základních školách. Na základě tohoto průzkumu mě oslovili někteří učitelé a vyjádřili zájem o mou práci. Těmto učitelům pošlu svou diplomovou práci přímo na email. Navíc jsem také přepracoval metodiku výuky do samostatného dokumentu a spolu s žákovskými listy je vystavil pro všechny učitele na portálu scratched.media.mit.edu. Dále plánuji vystavit svou práci také na nějakém českém webu podporujícím výuku.

Domnívám se, že programovací jazyk Scratch má velký potenciál a mohl by najít na základních školách velké uplatnění. Má velmi jednoduché a přehledné ovládání. Programování je tak pro začátečníky mnohem jednodušší. Sestavování programů je intuitivní a spočívá především ve skládání tzv. bloků příkazů, které do sebe zapadají podobně jako puzzle. Scratch je velmi dynamický a dovoluje měnit kód přímo za běhu programu a okamžitě sledovat výsledky těchto změn.

Na rozdíl od ostatních dětských programovacích prostředí je velkou výhodou programu Scratch, že je snadno dostupný, jeho plná verze je k dispozici veřejně na portálu scratch.mit.edu a především je zcela zdarma. Má velmi dobré interaktivní vlastnosti. Tím je vhodný také pro vytváření zajímavých multimediálních prezentací, animací a her.

Žáci mohou přímo v programu Scratch sdílet své projekty včetně zdrojového kódu a podělit se tak s ostatními o své aplikace. Díky tomuto přístupu mohou žáci také snadno převzít a modifikovat projekty ostatních uživatelů. Scratch tím podporuje učení se od ostatních uživatelů. To podle mého názoru velmi přispívá k rozvoji žáků a jejich motivaci.

Prostředí Scratch je velmi vhodné využít k výuce algoritmizace a základů programování na základních školách. Věřím, že v budoucnu by mohlo být programování součástí výuky informatiky na druhém stupni každé základní školy. Pro rozvoj myšlení žáků by to mělo jistě velký význam.

Pevně doufám, že svou metodikou a souborem úloh přispějí k zařazení základů programování do výuky informatiky. Budu také dále využívat prostředí Scratch při své výuce programování. Učím na střední škole a i přesto, že žáci jsou na Scratch „staří“, využijí toto prostředí v úvodu výuky algoritmizace a to především pro svou názornost (například příkazů *když* a *opakuj*).

Literatura a ostatní použité zdroje

- [1] RESNICK, M., Scratch.mit.edu [online]. 2. 7. 2009 [cit. 2012-04-21]. Getting Started Guide. Dostupné z WWW: <http://info.scratch.mit.edu/sites/infocratch.media.mit.edu/files/file/ScratchReferenceGuide14.pdf>
- [2] RESNICK, M., Scratch.mit.edu [online]. 13. 5. 2009 [cit. 2012-04-21]. Getting Started Guide. Dostupné z WWW: <http://info.scratch.mit.edu/sites/infocratch.media.mit.edu/files/file/ScratchGettingStarted14.pdf>
- [3] PECINOVSKÝ, J., Proč učit programování na základní škole. [online]. [cit. 2013-02-20]. Dostupné z: <http://www.ceskaskola.cz/2001/09/rudolf-pecinovsky-proc-ucit.html>
- [4] PECINOVSKÝ, J., Zásady správné výuky programování. [online]. [cit. 2013-02-20]. Dostupné z: <http://www.ceskaskola.cz/2001/09/rudolf-pecinovsky-zasady-spravne-vyuky.html>
- [5] BLAHO, A., KALAŠ, I., Imagine Logo Učebnice programování pro děti. Computer Press, Brno, 2006. ISBN 80-251-1015-X
- [6] VANÍČEK, J., MIKEŠ, R., Informatika pro základní školy a víceletá gymnázia 3. Díl. Computer Press, Brno, 2006. ISBN 80-251-1082-6
- [7] PECINOVSKÝ, J., Výuka objektově orientovaného programování žáků základních a středních škol. [online]. [cit. 2013-02-21]. Dostupné z: http://vyuka.pecinovsky.cz/prispevky/2003_OB_Vyuka_OOP_zaku_ZS_a_SS.pdf
- [8] MIKOLAJOVÁ, K., Konference DidInfo 2011. - Banská Bystrica: Univerzita Mateja Bela, 2011. - [nestr.]. - ISBN 978-80-557-0142-4 [DidInfo. Banská Bystrica, 7. - 8. 4. 2011] dostupné ve sborníku konference
- [9] PECINOVSKÝ, J., Současné trendy v metodice výuky programování. [online]. [cit. 2013-03-20]. Dostupné z: http://vyuka.pecinovsky.cz/prispevky/2006-PS_Soucasne_trendy_v_metodice_vyuky_programovani.pdf
- [10] HAWK, D., Learn Scratch. [online]. [cit. 2012-08-29]. Dostupné z: <https://mywebpace.wisc.edu/dhawk/scratch/>
- [11] WILSON, A., Introduction to Programming. [online]. [cit. 2012-11-24]. Dostupné z: <http://scratched.media.mit.edu/resources/introduction-programming>
- [12] CHIANG, J., Scratch Lessons: Shall We Learn Scratch Programming for Tweens. [online]. [cit. 2012-09-25]. Dostupné z: <http://scratched.media.mit.edu/resources/scratch-lessons-shall-we-learn-scratch-programming-tweens>
- [13] REDWARE, Resources for teachers, parents and computer club facilitators. [online]. [cit. 2012-10-13]. Dostupné z: <http://scratch.redware.com>
- [14] UNIVERZITA LA SALLE. LearnScratch. [online]. [cit. 2013-03-21]. Dostupné z: <http://learnscratch.org/>
- [15] BRENNAN, K., HAWSON, J., CHUNG, M., Scratch curriculum guide. [online]. [cit. 2012-06-25]. Dostupné z: <http://scratched.media.mit.edu/resources/scratch-curriculum-guide-draft>
- [16] HLAVATÁ, Eva. Tvořivá informatika s Baltíkem I. In: [online]. [cit. 2013-04-25]. Dostupné z: <http://tib.cz/tvorivyucitel/metodika.htm>

- [17] HLAVATÁ, Eva. Tvořivá informatika s Baltíkem II. In: [online]. [cit. 2013-04-25]. Dostupné z: <http://tib.cz/tvorivyucitel/metodika.htm>
- [18] HLAVATÁ, Eva. Tvořivá informatika s Baltíkem III. In: [online]. [cit. 2013-04-25]. Dostupné z: <http://tib.cz/tvorivyucitel/metodika.htm>
- [19] Imagine Logo. [online]. [cit. 2013-05-03]. Dostupné z: <http://www.pf.jcu.cz/imagine/>
- [20] KAČMÁŘ, Dalibor. *Jazyk C*. Vyd. 1. Praha, c2001. ISBN 80-722-6295-5
- [21] Internetový obchod knihy cpress. [online]. [cit. 2013-05-10]. Dostupné z: <http://knihy.cpress.cz/zakladni-skoly.html>
- [22] Imagine. [online]. [cit. 2013-05-10]. Dostupné z: <http://imagine.input.sk/cz/>
- [23] Scratch. [online]. [cit. 2013-05-10]. Dostupné z: <http://scratch.mit.edu>
- [24] Scratch 2.0. [online]. [cit. 2013-05-10]. Dostupné z: <http://scratch.mit.edu/projects/editor/>

Přílohy

Příloha č. 1: CD-ROM

- Diplomová práce ve formátu PDF
- Projekty ze studia pro učitele ve formátu sb2 zkomprimované do jednoho archivu zip
- Projekty ze studia pro žáky ve formátu sb2 zkomprimované do jednoho archivu zip

Příloha č. 2: Žákovské listy

Tyto listy jsou připraveny pro žáky a obsahují úkoly pro vývojové prostředí Scratch. Úlohy se mohou odkazovat na připravené projekty. Všechny tyto projekty jsou dostupné ze studia Výuka základů programování, na adrese <http://scratch.mit.edu/studios/408206/>

Projekty vytvořené v prostředí programu Scratch mohou být organizovány v tzv. Studiích. Jedná se o pojmenované uskupení projektů, které zpravidla mají něco společného, mají společný účel. Do studia lze přidávat vlastní projekty. Libovolné studio lze otevřít přes internetový odkaz nebo vyhledat podle názvu a tím si zobrazit všechny projekty, které obsahuje.

The screenshot shows the Scratch studio interface for 'Výuka základů programování'. At the top, there are navigation tabs for 'Projekty (11)', 'Komentáře (0)', 'Správci (1)', and 'Aktivita'. A 'Následovat' button is in the top right. Below the tabs, there is a 'Přidat projekty' button, a checkbox for 'Allow anyone to add projects', and a 'Třídění podle' dropdown menu. The main area displays a grid of 11 project thumbnails, each with a title and the creator's name 'od asjerk':

- 1 - První program
- 6 - Bludiště
- 6 - Vypínač
- 2 - Tanec postavy
- 2 - Mlsný kocour
- 3 - Pavouk
- 8 - Chytání švestek
- 7 - Fotbal - chyby
- 7 - Taneční párty - chyby
- 7 - Akvárium - chyby
- 5 - Kouzelná hůlka

Poznámka

Projekty umístěné ve studiu jsou pojmenované číslem (určuje téma, ke kterému projekt patří) a názvem úlohy. Například kapitola 5 obsahuje úlohu Kouzelná hůlka, proto připravený projekt se jmenuje 5 – Kouzelná hůlka.

1. Seznámení s prostředím Scratch

Scratch je programovací jazyk a online společenství kde můžeš tvořit vlastní interaktivní příběhy, hry, animace a všechny své výtvořky sdílet s ostatními lidmi z celého světa. Scratch je projektem Lifelong Kindergarten group na MIT Media Lab. Je dostupný zdarma na <http://scratch.mit.edu>

Seznámení s webem Scratch

Navštivte webové stránky <http://scratch.mit.edu/>. Procházením zjistíte, co všechno se na těchto webových stránkách nachází.

Procházení dostupných projektů

Prozkoumávejte web a zkuste zde nalézt zajímavé projekty. Nyní máte prostor, spustit si některé projekty. Pokud narazíte na nějaké hry, můžete si je klidně zahrát. Tlačítkem vpravo nahoře máte možnost pohlédnout do projektů a tím zjistit jak jsou projekty naprogramované.

Registrace na portálu

Zaregistrujte se nyní na stránkách <http://scratch.mit.edu>. Na těchto stránkách sdílejí uživatelé své projekty. Když se odkudkoliv přihlásíte na tento web, budete mít k dispozici prostředí programu Scratch a také všechny své projekty.

Poznámka

Program Scratch se nemusí instalovat a můžete s ním pracovat kdykoliv, odkudkoliv, dokonce i z domova.

Seznámení s prostředím editoru Scratch 2.0

Odkaz na prostředí programu Scratch 2.0 <http://scratch.mit.edu/projects/editor/>

Některé pracovní listy obsahují připravené projekty. Takový projekt si zobrazíte ze studia „Výuka základů programování“ na adrese <http://scratch.mit.edu/studios/408206/>. Poté stisknete tlačítko vpravo nahoře – **Pohlédni dovnitř**.

První program

Nejprve si otevřete projekt s názvem **1 – První program**, který naleznete v připraveném studiu. Tento program můžete otevřít také pomocí odkazu

<http://scratch.mit.edu/projects/21230742/#editor>.

Použijte příkazy, které leží na ploše, na níž se programuje. Příkaz spustíte tak, že na něj kliknete levým tlačítkem myši. Klikejte na jednotlivé příkazy a pozorujte nebo poslouchajte postavu v levé horní části okna.

Měňte číselné hodnoty u příkazů a pozorujte reakce postavy, když příkaz spustíte.

Na plochu můžete přetahovat další příkazy a vyzkoušet si je.

Příkazy můžete skládat do sebe podobně jako puzzle a vytvářet tak scénář. Takto složený scénář můžete spustit kliknutím, stejně jako jednotlivé příkazy.

O nové poznatky a příkazy se podělte s ostatními spolužáky.

2. Posloupnost příkazů a cyklus

Diskuze

Promyslete a následně vyjmenujte 5 situací ze života, ve kterých budete postupovat podle přesně definovaných kroků (instrukcí, pokynů nebo příkazů).

Tanec bez PC

Vytvořte dvojice. Jeden ze dvojice bude dávat instrukce druhému a ten je bude provádět.

Jedná se o jednoduchý tanec a pokyny znějí následovně:

„Dopředu - krok, krok, tleskni, dozadu – krok, krok, tleskni.“

Tyto instrukce opakujte a kamarád bude tancovat.

Instrukce, které udělíte kamarádovi, libovolně pozměňte a přidejte další.

Tanec postavy

Nejprve si otevřete projekt Tanec postavy, který naleznete v připraveném studiu na adrese <http://scratch.mit.edu/studios/408206/>

Na základě předchozího úkolu (Tanec bez PC) nyní naučte stejný tanec postavu ve Scratchi.

Postavě kočky nebo jiné udělejte povely tak, aby postava tančila ve vaší scéně. K tomu použijte příkazy, které leží na ploše, na níž se programuje.

Přijďte na to, jak taneční pohyb postavy 10krát zopakovat?

Až budete mít projekt hotov, můžete se s ním pochlubit ostatním například tak, že jej budete sdílet. Ostatní lidé jej budou moci prohlížet přes webové stránky scratch.mit.edu.

Mlsný kocour

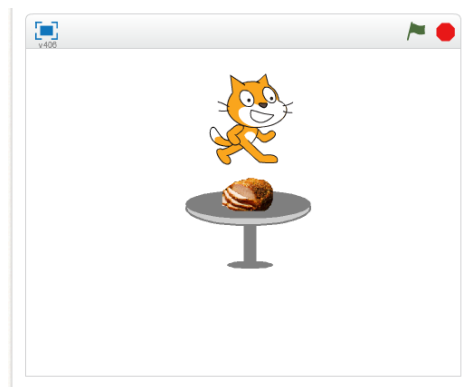
Využijte připraveného projektu „**Mlsný kocour**“ k naprogramování kocoura, který hladově obchází okolo stolu s pečínkou. Projekt je dostupný z připraveného studia

<http://scratch.mit.edu/studios/408206/>.

Projekt převezměte a podle svého upravte.

Poznámka

Proto, aby bylo možné projekt upravovat, nejprve musíte pohlédnout dovnitř a stisknout oranžové tlačítko umístěné vpravo nahoře „**Odvozené**“. Tím si projekt zkopírujete a uložíte k sobě do svých věcí a můžete na něm začít pracovat.



3. Postavy a pozadí

Kosmonaut na měsíci

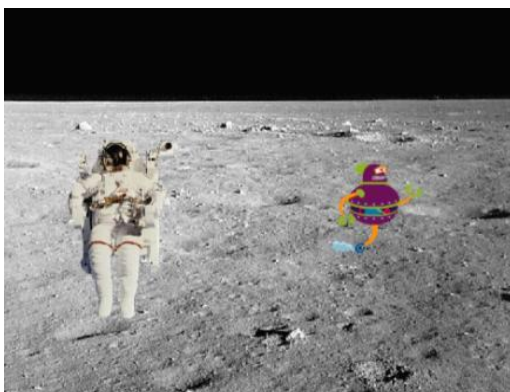
Otevřete si stránky Scratch z adresy <http://scratch.mit.edu/>. Přihlaste se pomocí odkazu vpravo nahoře. Vývojové prostředí spustíte odkazem vlevo nahoře (Tvořit). Nyní máte otevřen nový prázdný projekt. Doporučuji projekt pojmenovat. Pokud jste přihlášení, ukládat se bude automaticky.

Pokuste se změnit pozadí tak, aby scéna připomínala prostředí na Měsíci nebo ve Vesmíru podobně jako ukazuje obrázek vpravo.

Dokážete postavu kočky změnit na jinou postavu (kosmonauta)? Najděte (například na internetu) vhodný obrázek, který využijete pro změnu.

Postavu pak naprogramujte, ať se vznáší ve Vesmíru.

Pokuste se přidat do vesmírného prostoru ještě další objekt, který by tam mohl létat.



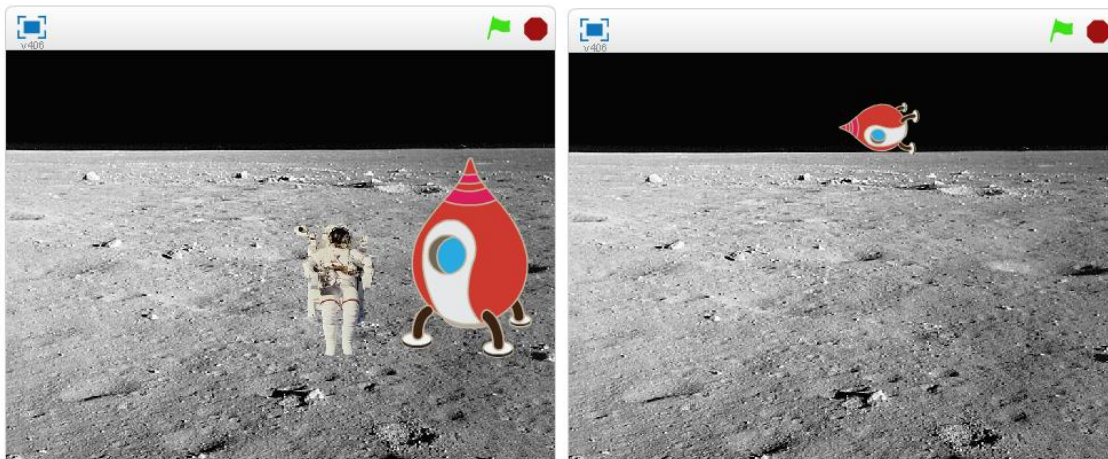
Kosmonaut na měsíci 2

Využijte již vytvořený projekt s kosmonautem, uložte si ho jako kopii a následně jej můžete libovolně upravovat, aniž by se úpravy týkaly původního projektu.

Vložte do scény postavu, která bude představovat raketu.

Scénář vytvořený pro kosmonauta upravte tak, aby doskákával k raketě a nastoupil.

Raketu naprogramujte k následnému odletu do Vesmíru.



Pavouk

Vaším úkolem v této úloze je postavit s pavoukem pavučinu.

Z připraveného studia si můžete odvodit projekt s názvem **Pavouk** (pavouka můžete také namalovat, nebo importovat obrázek).

V připraveném projektu je pouze postava pavouka bez scénáře. Projekt prozatím neobsahuje ani pozadí.

Nejprve změňte pozadí, vyberte pro scénu vhodný obrázek.

Postavu pavouka naprogramujte tak, aby při svém pohybu stavěl pavučinu.

Může pavouk za sebou zanechávat stopu (pavoučí vlákno)?

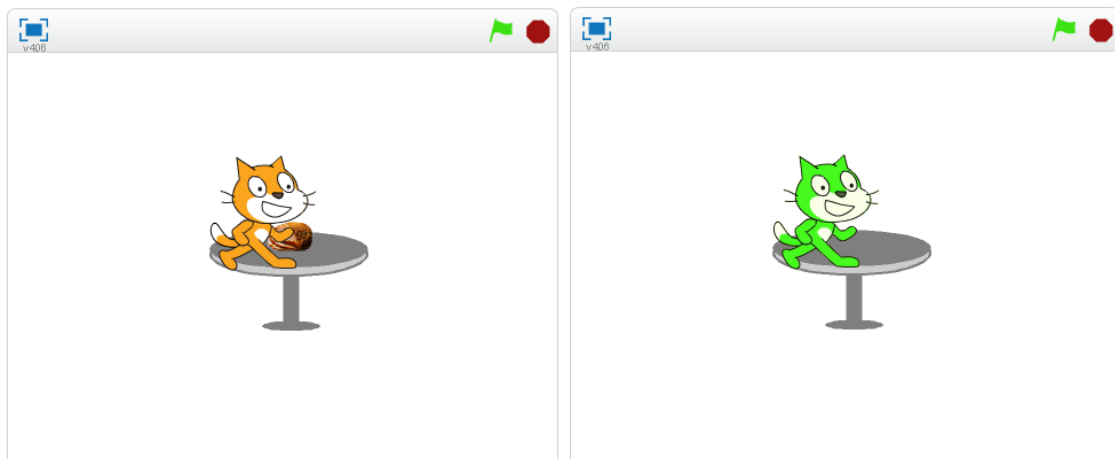


Mlsný kocour 2

Úlohu Mlsný kocour se pokuste upravit tak, aby kocour skočil na stůl a snědl pečínku.

Bude to i na něho dost velká porce a začne mu z toho být špatně. Dokážete ukázat, jak je z mastné pečínky kocourovi špatně?

Úpravy proveďte ve svém vlastním projektu nebo v připraveném projektu Mlsný kocour.



4. Více postav a postupný vývoj

Nežli se pustíte do programování úlohy, zodpovězte tyto otázky:

- Jaký je rozdíl mezi postavou a kostýmem postavy?
- Kdy použijete postavu?
- Kdy použijete kostým?

Taneční párty

Otevřete si program Scratch a v něm nový projekt.

Upravte pozadí scény, použijte vhodný obrázek z knihovny, nebo nějaký vlastní.

Do scény také přidejte libovolné postavy.

Postavy naprogramujte tak, aby reagovaly na kliknutí tancem.

Dokážete na párty přehrát svou oblíbenou píseň? Pokuste se vložit a přehrát ve scéně vlastní oblíbenou píseň.

Pracujte postupně po malých částech, vytvořenou část programu vždy otestujte. Pokud bude vše procovat správně, podle vašich představ, pokračujte dál ve vytváření scénáře.

Pokud je to možné, nafotěte se v různých tanečních pózách. Fotografie použijte ve svém projektu tak, že budete tančit na tanečním parketu.

Své práce můžete na závěr sdílet s ostatními.



5. Paralelizace a události

Animace akvária

Otevřete si program Scratch a v něm nový projekt.

Upravte pozadí scény, použijte vhodný obrázek z knihovny, nebo nějaký vlastní.

Do scény také přidejte libovolné postavy (vodní živočichy).

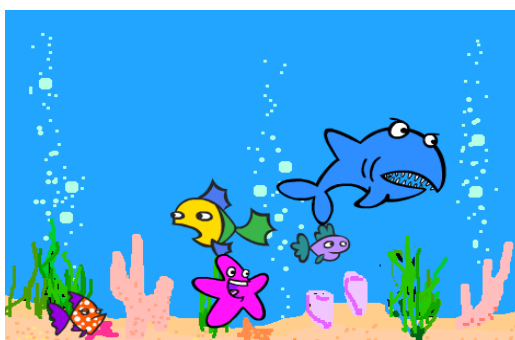
Postavy naprogramujte tak, aby samy plavaly v akváriu.

Přijďte na to, jak projekt naprogramovat, aby se všechny postavy začaly pohybovat na jediný povel – při kliknutí na zelenou vlajku?

Může se postava pohybovat neustále?

Co se stane, když narazí na okraj?

Lze zajistit, aby se postava při změně směru pohybu neotáčela (myšleno hlavou dolů)?



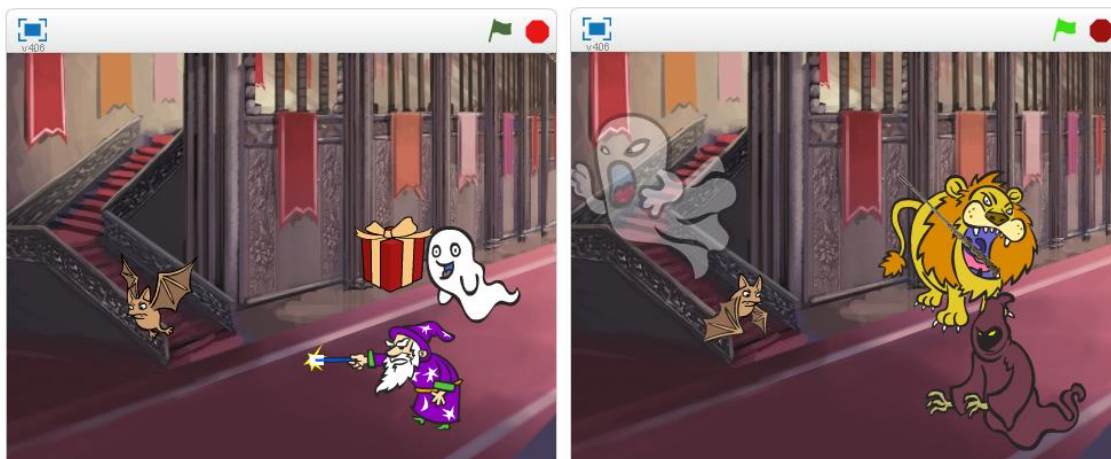
Kouzelná hůlka

Zahrajeme si na kouzelníka. Využijte připravený projekt s názvem **Kouzelná hůlka**, který naleznete v připraveném studiu <http://scratch.mit.edu/studios/408206/>.

Projekt obsahuje několik různých postav. Naprogramujte postavám pohyb, případně další efekty.

Připravenou kouzelnou hůlku naprogramujte tak, aby se pohybovala spolu s kurzorem myši.

Kliknutím na postavy budete kouzlit a postavy měnit v něco úplně jiného, nebo je necháte zmizet.



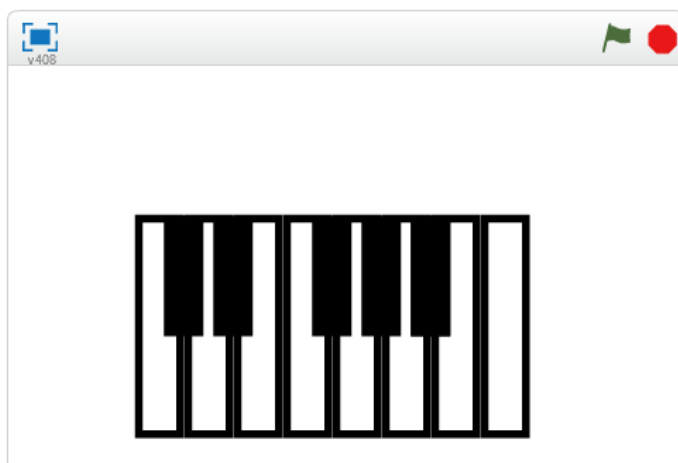
6. Události a zprávy

Hudební nástroj

Otevřete si editor Scratch a v něm nový projekt.

Vytvořte a naprogramujte hudební nástroj (například klavír), na který bude možné interaktivně hrát (klikáním myši).

Na internetu, nebo v hodině hudební výchovy najděte jednoduchou píseň a pokuste se ji přehrát na vašem nově vytvořeném nástroji.



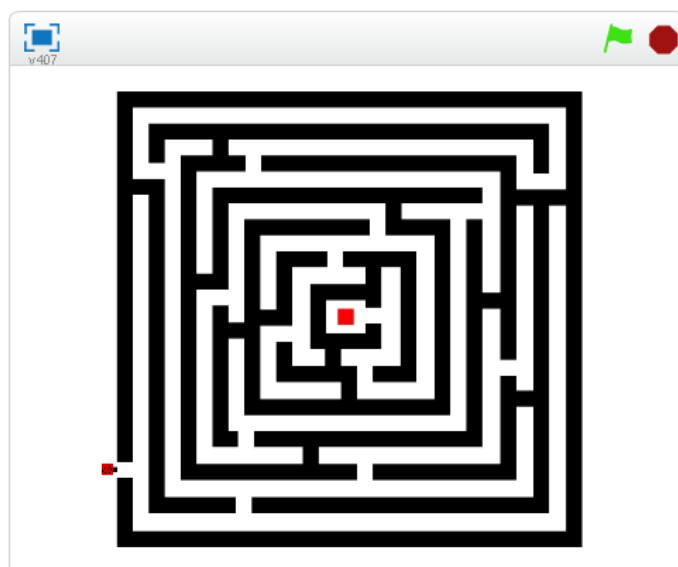
Bludiště

Otevřete si projekt **Bludiště** z připraveného studia.

Úkolem je využít události k dokončení rozpracovaného projektu.

V projektu je připravené pozadí (bludiště) a postava, která musí bludištěm projít.

Naprogramujte postavu tak, aby její pohyb byl možný ovládat pomocí šipek na klávesnici.



Vypínač

Otevřete si projekt **Vypínač** z připraveného studia.

Úkolem je využít události – zprávy k dokončení rozpracovaného projektu.

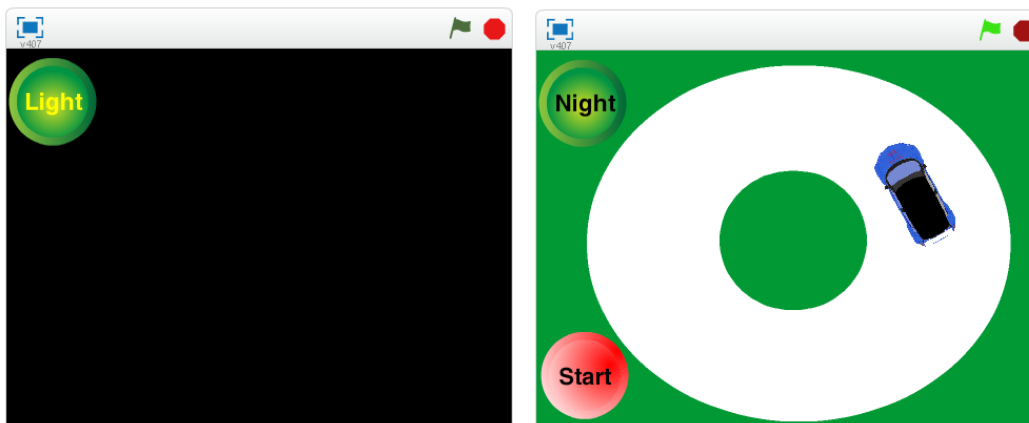
V projektu máte připravené pozadí, tlačítka i automobil.

Při spuštění projektu chceme, aby byla tma a bylo viditelné pouze tlačítko s nápisem Light. Po stisku tohoto tlačítka se nám rozední a ukáže se nám pozadí dráhy, automobil a tlačítko Start. Stiskem stejného tlačítka bychom měli mít možnost kdykoliv zhasnout světlo.

Stiskem tlačítka Start dostane automobil povel a začne se pohybovat po dráze.

K naprogramování komunikace mezi postavami navzájem a scénou využijte zprávy.

Každá postava má více kostýmů, které využijte k dosažení nejlepšího výsledku.



Razítka

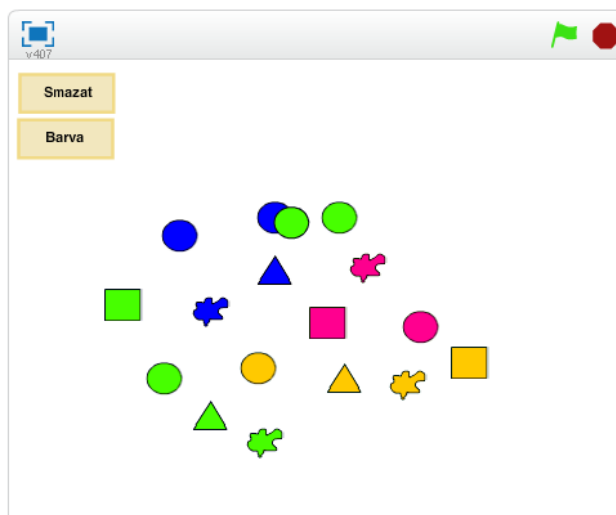
Otevřete si editor Scratch a v něm nový projekt.

Vaším úkolem je nakreslit postavu (razítka). Pro razítka vytvořte více tvarů (kostýmů), které bude možné přepínat například stisknutím klávesy mezerník.

Razítka bude přichycené k myši a na kliknutí se otiskne do scény.

Vytvořte tlačítko, pomocí kterého bude možné měnit barvy razítka.

Vytvořte ještě jedno tlačítko, které bude moci vymazat plochu.



7. Testování a ladění

Otevřete si postupně všechna připravená cvičení. Vaším úkolem je v těchto programech nalézt chyby. Poté navrhnout způsob odladění a programy odladit.

Taneční párty – chyby

Otevřete si projekt **Taneční párty – chyby** z připraveného studia.

Úloha obsahuje čtyři postavy.

Programátor v projektu udělal chyby nebo nesprávně použil některé příkazy.

Chyby nalezněte a opravte.



Akvárium – chyby

Otevřete si projekt **Akvárium – chyby** z připraveného studia.

Tento projekt obsahuje pět postav – vodních živočichů a pozadí s mořským dnem. Při spuštění projektu můžete na první pohled zjistit, že čtyři ryby nefungují správně.

Program obsahuje chyby a nefunguje tak, jak si autor představoval. Projekt překontrolujte a opravte případné nedostatky.



Fotbal – chyby

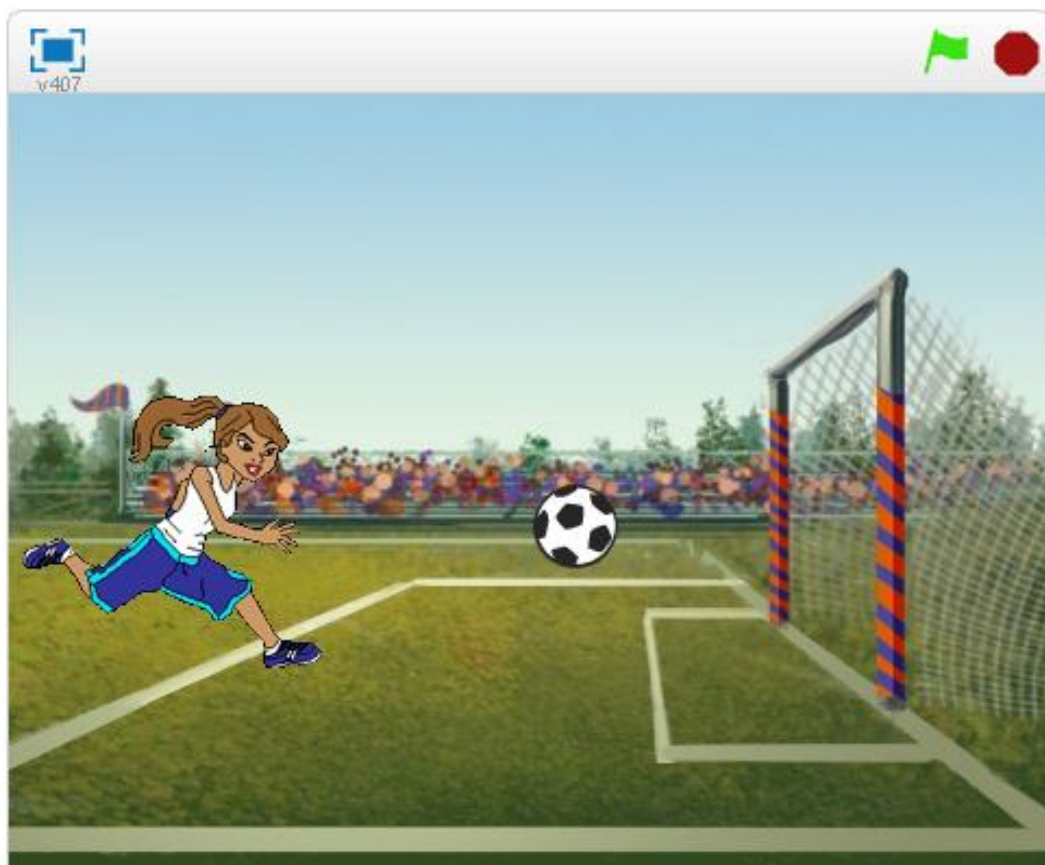
Otevřete si projekt **Fotbal – chyby** z připraveného studia.

Tento projekt obsahuje postavu dívky, míč a pozadí s fotbalovou brankou.

Dívka by měla přiběhnout k míči, poté uživatel stiskne klávesu mezerník a v ten okamžik by měl míč letět do branky. Následně by měla dívka jásat.

Program ovšem tak nefunguje.

Vaším úkolem je chybu najít a opravit.



8. Klonování

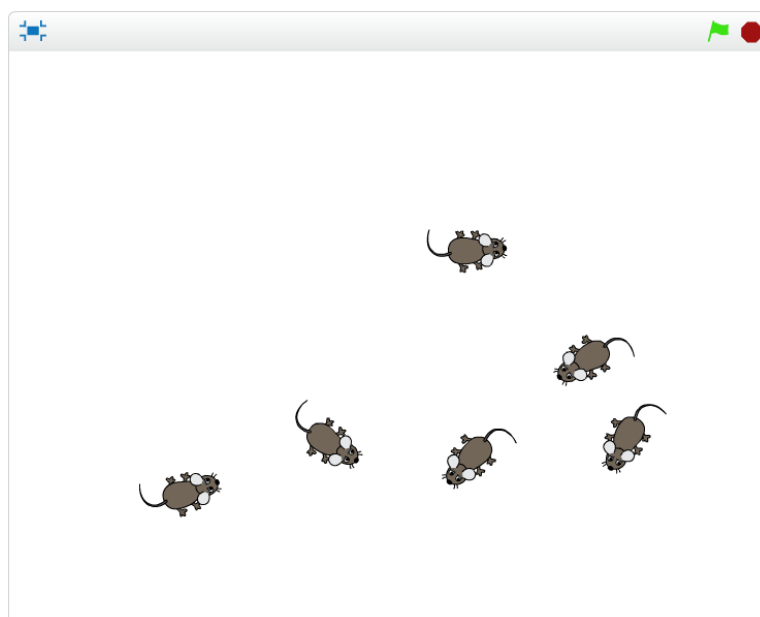
Myši

Otevřete si editor Scratch a v něm nový projekt.

Váším úkolem je vytvořit program, kde se budou množit myši.

Myši se budou náhodně objevovat a budou běhat po scéně.

Nejprve vložte do scény postavu (myš). Tuto postavu budete klonovat. Vznikající klony se budou náhodně pohybovat po scéně.



Možnosti rozšíření programu

Následně můžete program upravit tak, aby bylo možné myš odstranit kliknutím na ni.

Také můžete zkusit program rozšířit o další postavu – kočku, která bude myši chytat.

Arkanoid – příprava hry

Otevřete si editor Scratch a v něm nový projekt.

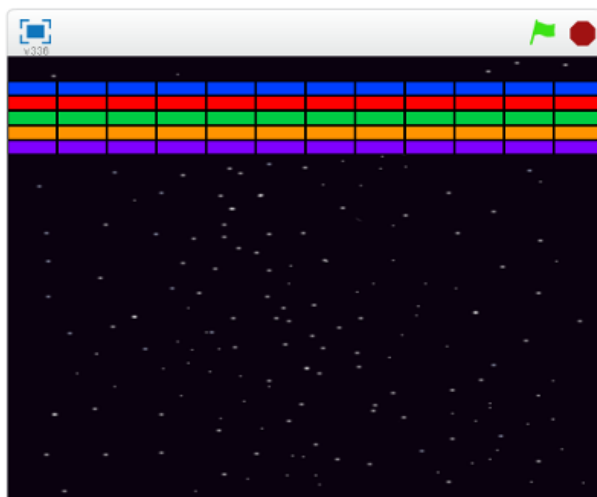
Vaším úkolem je vytvořit základ pro hru Arkanoid.

Do scény nejprve vložte vhodné pozadí.

Připravte si jednu cihličku (postavu) a tu posléze naklonujte na vhodné pozice ve scéně.

Udělejte nejméně pět řad a každou řadu jinou barvou podobně, jak ukazuje obrázek.

V následujících tématech budeme projekt dále rozvíjet.



Chytání švestek – příprava hry

Otevřete si projekt **Chytání švestek** z připraveného studia.

Projekt obsahuje pozadí a tři postavy – rytíře, košík a švestku.

Vaším úkolem je upravit tento připravený projekt tak, aby se v horní části scény náhodně objevovaly švestky a po nějaké době padaly dolů na zem. Využijte postavu švestka, pro kterou vytvoříte vhodný scénář.

Postavy v dolní části scény zatím nijak neupravujte, později je naprogramujete k chytání švestek do košíku.



9. Podmíněný příkaz a operátory

Chytání švestek

Otevřete si rozpracovaný projekt z předchozího tématu.

Padající švestky bychom nyní potřebovali chytat do košíku.

Pokud se padající švestka dotkne košíku, košík ji do sebe pojme a švestka zmizí.

Využijte podmíněný příkaz „když“ a příkazy vnímání k dopracování tohoto efektu.

Později k projektu přidáme počítání chyčených švestek.



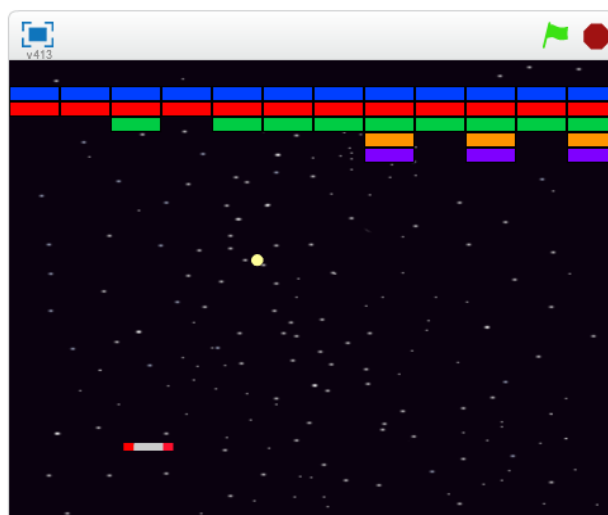
Arkanoid

Otevřete si projekt **Arkanoid**, který máte rozpracovaný z předchozího tématu.

Máme připravené pouze naklonované cihličky v horní části scény.

Vaším úkolem je přidat do scény kuličku spolu s tyčkou a naprogramovat jejich scénáře.

Kulička se bude odrážet od tyčky, okrajů scény a také od barevných cihliček, které bude zároveň bořit (cihličky budou mizet).



10. Proměnné

Chytání švestek – skóre a životy

Otevřete si projekt **Chytání švestek**, který máte rozpracovaný z předchozího tématu.

Nyní rozšíříme tento projekt o počítání chycených švestek do košíku a také omezíme životy (kolik švestek nám může spadnout na zem). V případě že hráč vyčerpá všechny životy, hra bude ukončena.

Skóre i životy se pokuste naprogramovat pomocí proměnných.



Arkanoid – skóre a životy

Otevřete si projekt **Arkanoid**, který máte rozpracovaný z předchozího tématu.

Tuto hru také vylepšete s pomocí proměnných o skóre (počítání zasažených cihliček) a omezte počet životů.

Pokuste se hru rozšířit například o řízení rychlosti kuličky. Kulička by mohla v závislosti na skóre zrychlovat.

V následujícím tématu naprogramujete postup do dalšího levelu.



Kalkulačka

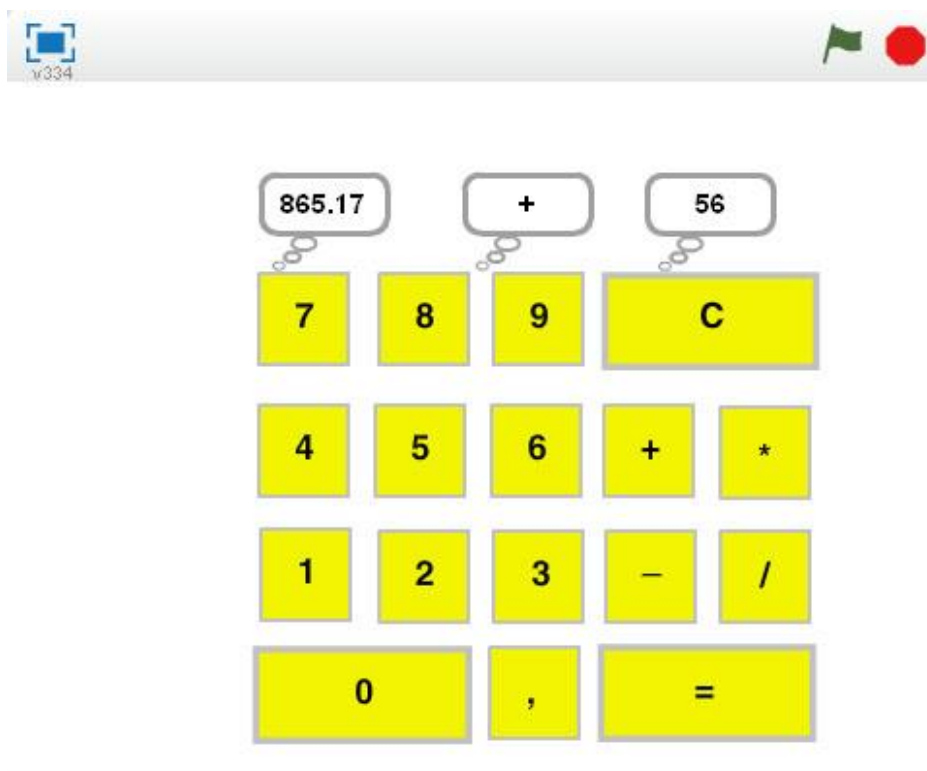
Otevřete si editor Scratch a v něm nový projekt.

S pomocí proměnných vytvořte program, který bude fungovat jako jednoduchá kalkulačka.

Kalkulačka bude umět sčítat, odečítat, násobit a dělit.



Vytvořte jednotlivá tlačítka kalkulačky, na která bude možné klikat myší a tím zadávat hodnoty, případně operace s čísly.

Pro uchování zadávaných čísel použijte seznam.



11. Procedury

Poznámka

Procedura je pojmenovaná část skriptu. Vytvoříme ji za pomoci bloku . Volání procedury je možné pomocí bloku . Procedury má smysl používat v programech, kde se část příkazů vyskytuje vícekrát a také pro přehlednost zdrojového kódu.

Arkanoid – další levely

Otevřete si projekt **Arkanoid**, který máte rozpracovaný z předchozího tématu.

Najděte ve vytvořené hře část příkazů, které se vyskytují v programu opakovaně, nebo část která provádí určitou činnost, která by stála za to pojmenovat a tím zpřehlednit zdrojový kód.

Z nalezené části vytvořte proceduru a tu potom budete volat v programu.

Ve hře může být také implementována procedura, která bude zajišťovat postup do vyšší úrovně.



Závod

Otevřete si editor Scratch a v něm nový projekt.

Vytvořte závodní hru, ve které využijete procedury.

Tratí závodu může být např. jednoduchý ovál.

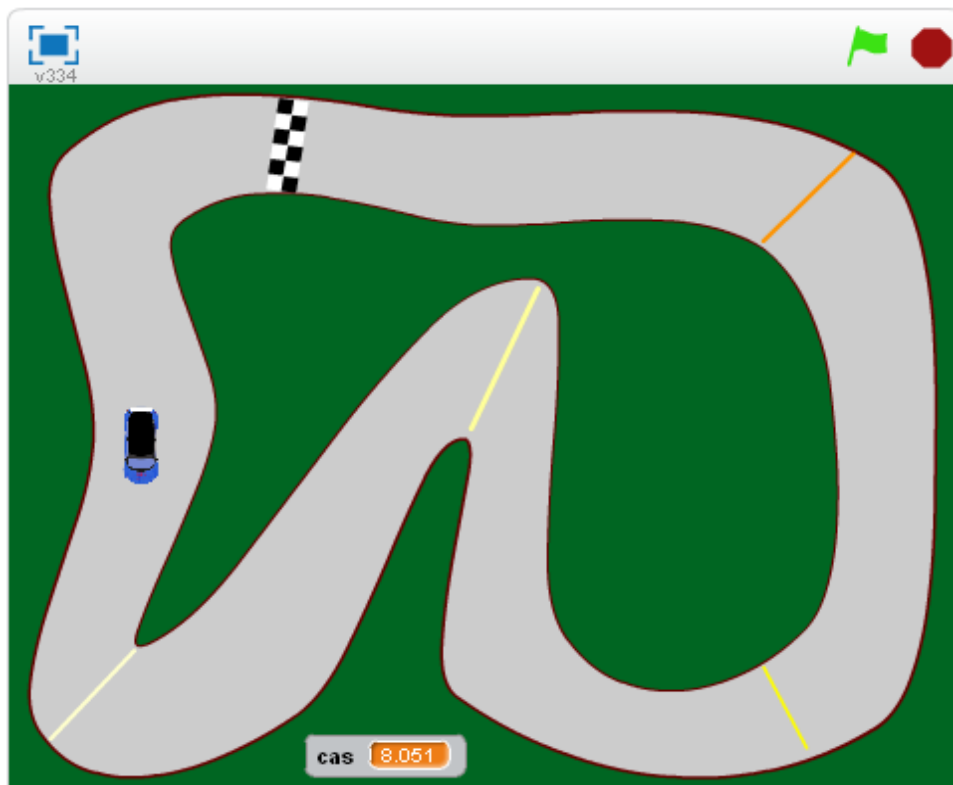
Uživateli umožněte ovládání závodního auta pomocí klávesnice, auto může i zrychlovat a zpomalovat.

Dokážete změřit čas, po který bude závodník na trati?

Lze uchovávat rekordní čas?

Hru následně můžete vylepšovat například o kontrolní průjezdní body, nebo zajistit zpomalení auta na ploše mimo silnici.

Příkladem použití procedury můžete být právě zajištění této kontroly – průjezdních bodů, auta mimo silnici, nebo dojezd do cíle.



12. Závěrečný projekt

Zadání

Vaším úkolem je zpracovat rozsáhlejší projekt, například vlastní hru.

Nejprve si rozmyslete, na jaké téma budete projekt zpracovávat a následně si zpracujte scénář projektu. Až budete mít vše dobře rozmyšlené, můžete se pustit do vytváření grafiky a programování.

Příkladem závěrečného projektu mohou být například hry:

