

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA PEDAGOGICKÁ
KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

MOŽNOSTI VYUŽITÍ PROGRAMU SCRATCH
BAKALÁŘSKÁ PRÁCE

Milan Kryl

Informatika se zaměřením na vzdělávání

Vedoucí práce: Mgr. Tomáš Přibáň, Ph.D.

Plzeň, 2015

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně s použitím uvedené literatury a zdrojů informací.

V Plzni, 30. června 2015

.....
vlastnoruční podpis

Poděkování

Rád bych tímto poděkoval vedoucímu bakalářské práce Mgr. Tomáši Přibáňovi, Ph.D. za ochotu, vstřícnost při konzultacích a věcné rady při zpracování mé bakalářské práce.

ZDE SE NACHÁZÍ ORIGINAL ZADÁNÍ KVALIFIKAČNÍ PRÁCE.

OBSAH

SEZNAM ZKRATEK	3
ÚVOD	4
1 PŘEDSTAVENÍ PROGRAMU SCRATCH.....	5
1.1 CO JE SCRATCH.....	5
1.2 HISTORIE.....	6
1.2.1 Scratch 1.0	6
1.2.2 Scratch 1.4	7
1.3 AKTUÁLNÍ VERZE.....	9
1.3.1 Scratch 2.0 online editor	9
1.3.2 Scratch 2.0 offline editor	10
1.3.3 Instalace.....	10
1.3.4 Popis prostředí Scratch 2.0.....	12
1.3.4.1 Hlavní Menu	13
1.3.4.2 Paleta Nástrojů	14
1.3.4.3 Skriptovací oblast	15
1.3.4.4 Paleta bloků	17
1.3.4.4.1 Scénáře	17
1.3.4.4.2 Kostýmy - Pozadí	19
1.3.4.4.3 Zvuky.....	20
1.3.4.5 Scéna, Seznam postav	20
1.3.4.5.1 Scéna	20
1.3.4.5.2 Seznam postav.....	20
1.3.5 Porovnání Online a Offline editoru.....	22
1.4 MOŽNOSTI VYUŽITÍ PROGRAMU	22
1.4.1 Výuka základů programování	22
1.4.2 Tvorba interaktivních výukových prezentací.....	23
1.4.3 Scratch a OOP	25
2 UŽIVATELSKÁ KOMUNITA	27
2.1 SCRATCH ÚČET	27
2.1.1 Vytvoření Scratch účtu	27
2.2 SCRATCH WIKI	27
2.3 DISKUZNÍ FÓRA.....	28
2.3.1 Welcome to Scratch	28
2.3.2 Making Scratch Projects	28
2.3.3 About Scratch	29
2.3.4 Interests Beyond Scratch.....	29
2.3.5 Scratch in Other Languages	29
2.4 SCRATCHED.....	30
2.5 SCRATCHJR.....	30
3 MOŽNOSTI VYUŽITÍ – UKÁZKA PRAKTICKÝCH PŘÍKLADŮ	32
3.1 PRVNÍ PROJEKT.....	32
3.1.1 Cíle	32
3.1.2 Popis řešení	32
3.2 TŘÍDĚNÍ ODPADU	32
3.2.1 Cíle	32
3.2.2 Popis řešení	32

3.3	BLUDIŠTĚ.....	33
3.3.1	Cíle	33
3.3.2	Popis řešení	33
3.4	SKÁKÁNÍ NA PLATFORMY	35
3.4.1	Cíle	35
3.4.2	Popis řešení	35
3.5	HRA STŘÍLENÍ NA NETOPÝRY	36
3.5.1	Cíle	36
3.5.2	Návrh řešení	36
3.6	ANIMACE	38
3.6.1	První projekt	38
3.6.2	Přidávání postav, pozicování, Interakce, posílání zpráv	38
3.6.3	Tvorba bloků	38
	ZÁVĚR.....	39
	RESUMÉ	40
	SEZNAM LITERATURY	41
	SEZNAM OBRÁZKŮ, TABULEK, GRAFŮ A DIAGRAMŮ	43
	PŘÍLOHY	I

SEZNAM ZKRATEK

MAC – Macintosh

OOP – Objektově orientované programování

MIT – Massachusettský technologický institut

ZTGI – Zpracování technických a grafických informací (název předmětu)

Úvod

Dnešní svět je přesycený elektronikou, počítači a všemožnými „smart“ zařízeními, které nás obklopují v našich každodenních činnostech. S přihlédnutím k tomuto faktu jsou kladeny stále větší nároky na počítačovou a technickou gramotnost. Na druhou stranu jsou v dnešní době již dostupné různé produkty, které počítačovou gramotnost pomáhají zvyšovat. Pokud před někým, kdo se o nové technologie v oblasti výpočetní techniky příliš nezajímá, vyslovíte slovo *programování* nebo *algoritmus*, jeho reakce jsou většinou smíšené a mnohým z nich představa, že by měli programovat, přijde nereálná. Hlavním důvodem bývá strach z neznámého. Lidé si pod programováním a algoritmizací většinou představí psaní sáhodlouhých kódů se složitou syntaxí.

Mnozí z nich však nevědí, že existují vizuální programovací jazyky jako je například Scratch. Troufám si říct, že programování v něm je zábavné. Naprosto odpadá jakákoliv nutnost znalosti syntaxe, protože programování ve Scratchi probíhá pomocí přesouvání příkazů ve formě barvených bloků, které do sebe zapadají jako jakési pseudopuzzle. Pokud si k tomuto přidáme navíc pěkné programovací prostředí, barevné postavy, kterým se pomocí bloků definuje jejich chování, interakce s ostatními postavami nebo např. reakce na stisknuté klávesy, tak lze tvrdit, že programování může být zábavné.

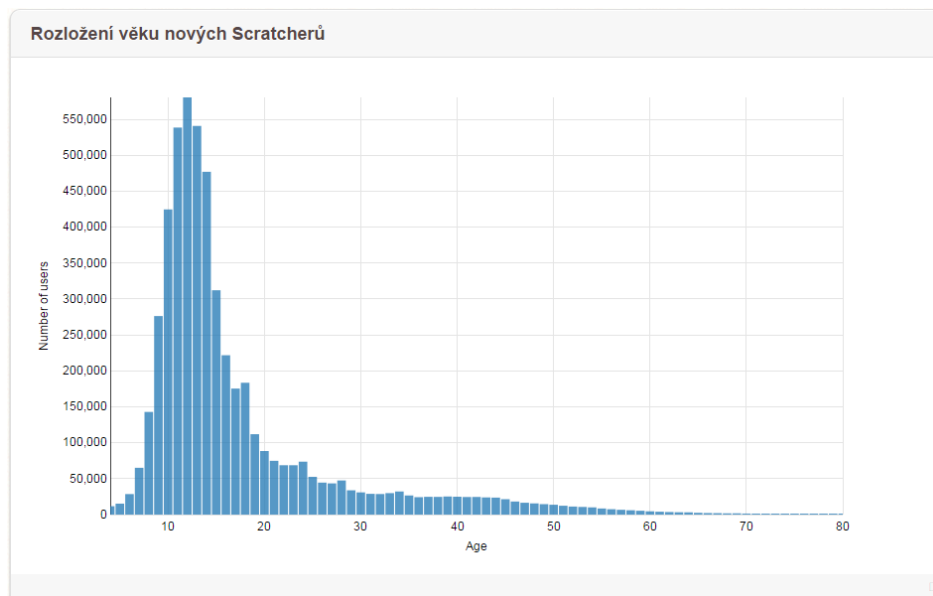
Cílem této práce je představit programovací prostředí Scratch a s pomocí několika příkladů popsat možnosti jeho využití. Téma jsem zvolil z toho důvodu, že jsem se sám učil programovat v objektově orientovaném programovacím jazyce (Java) a musím říct, že začátky byly velmi obtížné, velké problémy mi činila syntaxe jazyka a absence okamžitě viditelného výsledku. Scratch oba tyto problémy eliminuje a vybral jsem si ho proto, že neustále roste počet jeho uživatelů.

1 PŘEDSTAVENÍ PROGRAMU SCRATCH

1.1 CO JE SCRATCH

Scratch je programovací jazyk vyvinutý k výuce programování a rozvoji programátorského myšlení. Programování ve Scratchi probíhá pomocí přesouvání příkazů na skriptovací plochu, odpadá tedy nutnost znát syntaxi jako je tomu u jiných jazyků, což představuje velkou výhodu pro začínající programátory. Bloky jsou barevně odděleny, podle toho, jakou oblast příkazů reprezentují a jmenovitě to jsou: *Pohyb, vzhled, zvuk, pero, data, události, ovládání, vnímání, operátory a bloky*. Všem oblastem se bude práce věnovat v další části a podrobně je bude i analyzovat.

Scratch svým pojetím cílí především na žáky druhého stupně základní školy. Jak je možné vidět z grafu na obrázku 1, největší skupinu zastupují uživatelé ve věku 10 až 14 let, přičemž nejpočetnější zastoupení mají uživatelé ve věku 12 let.



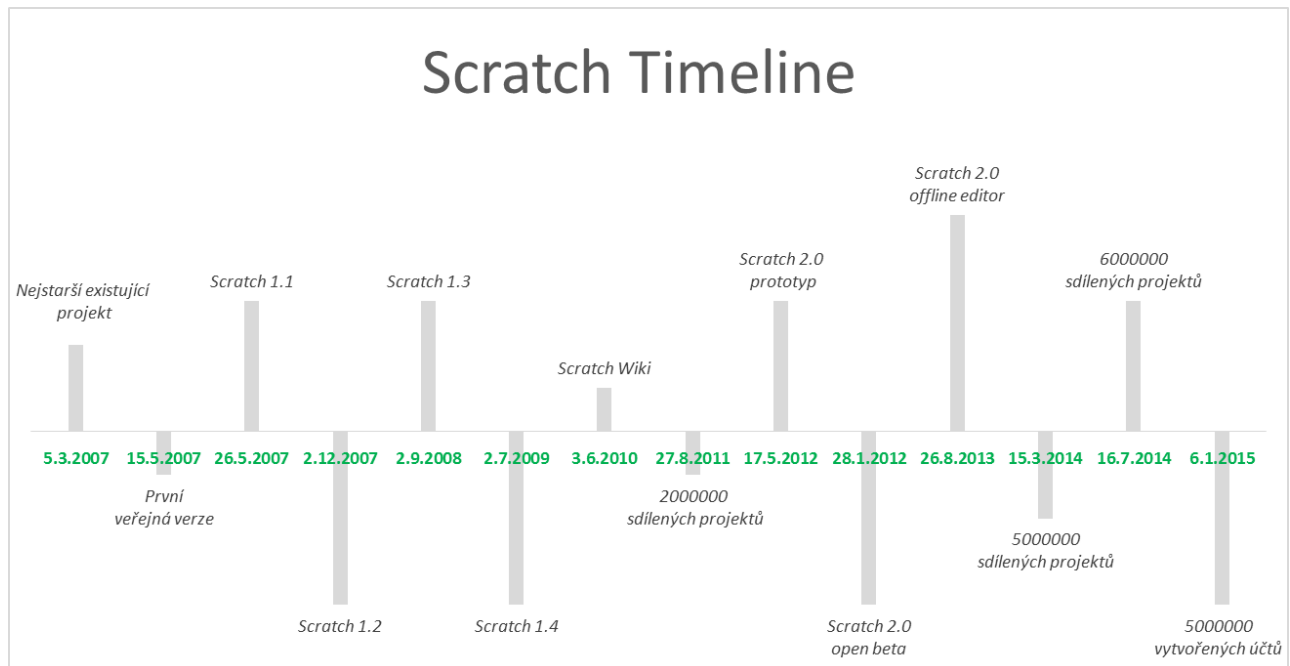
Obrázek 1: Rozložení věku nových uživatelů [1]

Další, méně početnou skupinu tvoří uživatelé v rozmezí 15 až 20 let. Počet nových uživatelů poté s narůstajícím věkem pozvolna klesá. Z toho vyplývá, že Scratch není určen pouze dětem a náctiletým, ale využívají ho i dospělí, mnoho z nich jsou právě učitelé programování, kteří Scratch používají k výuce nebo tvorbě studijních materiálů a prezentací.

1.2 HISTORIE

Historie programovacího prostředí Scratch se začala psát v roce 2003, kdy Mitch Resnick, John Maeda a Yasmin Kafai podali návrh na vytvoření vizuálního programovacího jazyka a na základě přiděleného grantu začali pracovat na verzi 0.1. Verze se od nynější lišila hlavně rozložením a také odlišným vzhledem bloků, které byly více hranaté [2].

V následujícím grafu jsou zaznamenány nejvýznamnější milníky z historie Scratche [3].



Obrázek 2: Scratch Timeline

1.2.1 SCRATCH 1.0

Scratch 1.0 byl vydán 8. ledna 2007, jednalo se o jeho první veřejnou verzi programu. Obsahoval celkem 95 bloků. Bloky se nepřesouvaly na skriptovací plochu jako je tomu u aktuální verze (verze 2.0), ale na skriptovací plochu se dostaly poklepáním na příslušný blok. Tato verze neobsahovala ani možnost přidávat komentáře. Kromě základních bloků obsahovala navíc experimentální bloky, které bylo možné zpřístupnit v položce menu *Extras* [4].

Experimentální bloky nefungovaly dobře a tím pádem se neobjevovaly v paletě příkazů. Mezi tyto bloky patřilo 13 zvukových bloků, které byly používány s nadstavbou programu Scratch. Tato nadstavba umožňovala přehrávání videí v programu.

Mezi experimentální bloky se dále řadí vjemové bloky a kontrolní bloky [5].

1.2.2 SCRATCH 1.4

Verze 1.4 byla poslední verzí první série a je stále k dispozici ke stažení. Na rozdíl od ostatních verzí první série obsahovala mnoho zásadních změn a vylepšení uživatelského prostředí a také některých bloků. Byly přidány motorové bloky pro použití se stavebnicí LEGO WeDo [6].

Motorové bloky lze ve Scratchi 1.4 zobrazit po kliknutí na položku menu *Editovat* a poté kliknutím na položku *Zobrazit motorové bloky*. Po tomto kroku jsou již bloky viditelné a lze s nimi pracovat. Jsou k nalezení v částech *Pohyb* a *Vnímání*. V pohybové sekci se konkrétně jedná o bloky: *Zapni motor na x sekund*, kde *x* je počet sekund zadávaný uživatelem, *Zapni motor*, *Vypni motor*, bloky pro nastavení síly motoru a poslední blok kterým lze měnit směr otáčení motoru. Sekce *vnímání* je na motorové bloky o něco chudší, obsahuje bloky pouze dva. Jedná se o jednu proměnnou, která uchovává hodnotu senzoru, druhým blokem je booleanovský blok uchovávající logickou hodnotu o stisknutí tlačítka nebo propojení senzorů A až B [7].

S verzí 1.4 se váží i některé modifikace programu Scratch. Jedná se o upravené verze programu. Důvodem k jejich vzniku bylo doplnění stávající verze programu o některé další nedostupné funkce nebo bloky. Tyto upravené verze nebyly podporovány originální verzí programu a projekty v nich vytvořené nemohou být sdíleny na webu [8] [9].

Těchto modifikací existuje celá řada, vybral jsem dva zástupce, které krátce představím. První z modifikací, kterou jsem ji vybral, je *Bingo*. Tato modifikace upravuje uživatelské rozhraní a přidává či upravuje přibližně 80 bloků. Zajímavou možností je export projektu do souboru s příponou **.exe* [10].

Druhou zdařilou modifikací, kterou jsem si vybral je *BYOB (Build Your Own Blocks)*, který je nyní známý jako *Snap!*.

Modifikace *Build Your Own Blocks* umožňovala vytvoření vlastních bloků, klonování a s ním spojenou hierarchii postav, kompilátor (vytvoření **.exe* nebo **.app* souboru), debugger, možnost vytvářet vícerozměrné seznamy (pseudopole).

V roce 2009 se univerzita v Berkeley rozhodla vytvořit programovací jazyk, který bude ze Scratche vycházet, ale bude zároveň schopný pomoci pochopit nejenom jednoduché cykly,

podmínky apod. vznikl tak *Build Your Own Blocks* verze 4.0, který byl posléze přejmenován na *Snap!*.

Snap! cílí na jinou věkovou kategorii než Scratch. Cílovou kategorií jsou studenti středních škol, kteří se s programováním seznamují a mohlo by je kreslené prostředí Scratche, které je určeno především mladším kategoriím, odradit.

Podobně jako ve *Scratch 2.0 online editor*, probíhá práce v programovacím jazyce *Snap!* v internetovém prohlížeči. Celé prostředí je implementováno pomocí JavaScriptu. *Snap!* je dostupný na webové adrese <http://snap.berkeley.edu/snapsource/snap.html>. Myslím, že by vývojáři mohli zvážit offline verzi programu, důvodem je občasné zamrznutí celého prostředí, stejně jako je tomu při práci ve *Scratch 2.0 online editoru* [11][12].

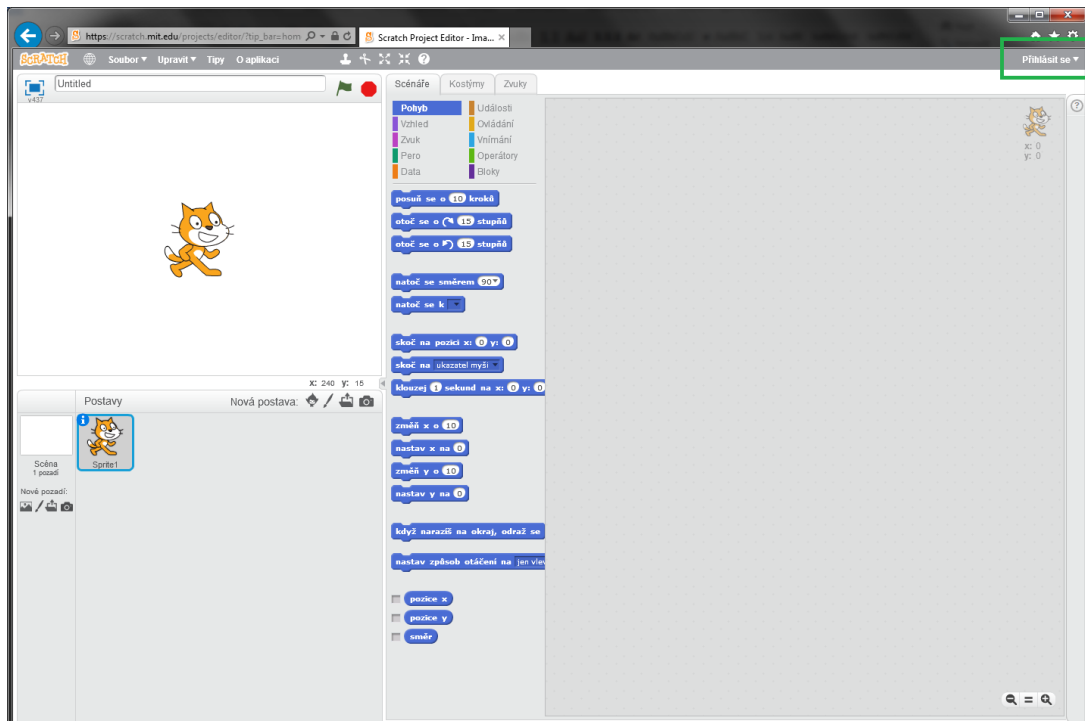
1.3 AKTUÁLNÍ VERZE

1.3.1 SCRATCH 2.0 ONLINE EDITOR

Verze 2.0 se poprvé objevila 28. ledna roku 2013, kdy se jednalo o beta verzi. Plná verze byla oficiálně zpřístupněna o něco později téhož roku, konkrétně se jednalo o 9. května.

S novou verzí přišlo i mnoho vylepšení. Přibyl blok *Název pozadí, Batoh*, který je určený pro přenos postav, kostýmů a skriptů mezi projekty. Nově je umožněno klonování postav až do počtu 300 klonů, dále je možné vytvářet cloudové proměnné, vlastní bloky kódu, jejichž vytváření by se dalo přirovnat k tvorbě metod. Je možné upravovat svoji uživatelskou stránku na webu scratch.mit.edu, přibyl zvukový editor, časové bloky, při kreslení kostýmů je nově možnost použít i vektorový editor.

Scratch online editor je dostupný na stránkách scratch.mit.edu, kde se po kliknutí na odkaz *Tvořit* dostanete přímo do editoru, kde můžete okamžitě začít vyvířet projekty. Prostředí online editoru zprostředkovává stejnou zkušenost jako offline editor. Nabízí však navíc možnost přihlásit se prostřednictvím Scratch účtu po kliknutí na odkaz v pravém horním rohu aplikace viz Obrázek 3.



Obrázek 3: Scratch 2.0 online editor

Po přihlášení máte možnost ukládat své vytvořené projekty do cloudového úložiště. To umožňuje pracovat na projektech téměř z jakéhokoli počítače s přístupem k internetu [13].

1.3.2 SCRATCH 2.0 OFFLINE EDITOR

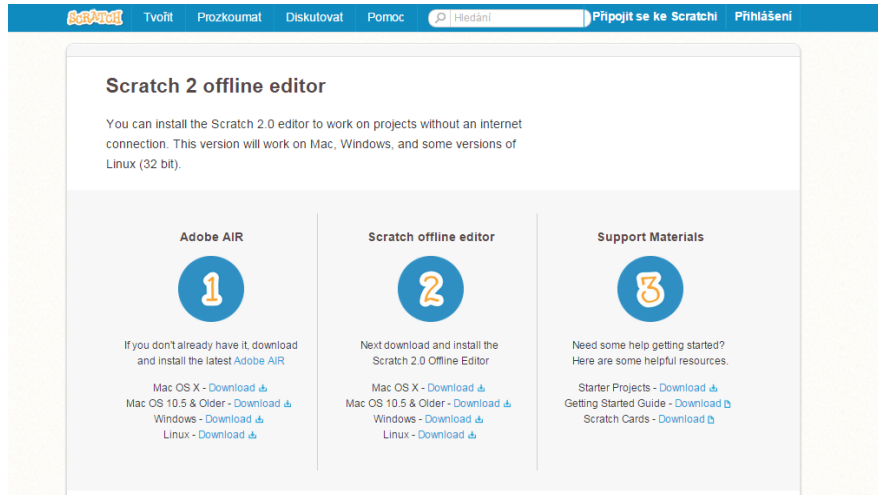
Scratch 2.0 offline editor je verze Scratche 2.0, která může být stažena do počítače, na rozdíl od online Flash editoru. Pokud bychom se zaměřili na systémové požadavky, nejsou nikterak náročné, postačí 32bitový operační systém Windows, MAC nebo Linux. Dále Adobe Air verze 2.6 a vyšší a přes 23MB volného místa na pevném disku.

V Offline editoru verze 2.0 je možné otevírat i projekty vytvořené v předchozích verzích, které projekty ukládaly s příponou *.sb. Verze 2.0 projekty ukládá s příponou *.sb2. Projekty uložené ve verzi 2.0 nelze bez pomoci externího programu otevřít v předchozích verzích.

Verze 2.0 umožňuje ukládat hodnoty proměnných v cloudu. Tato možnost je hojně využívána u některých vytvořených her. Například k ukládání nejlepšího dosaženého skóre [14].

1.3.3 INSTALACE

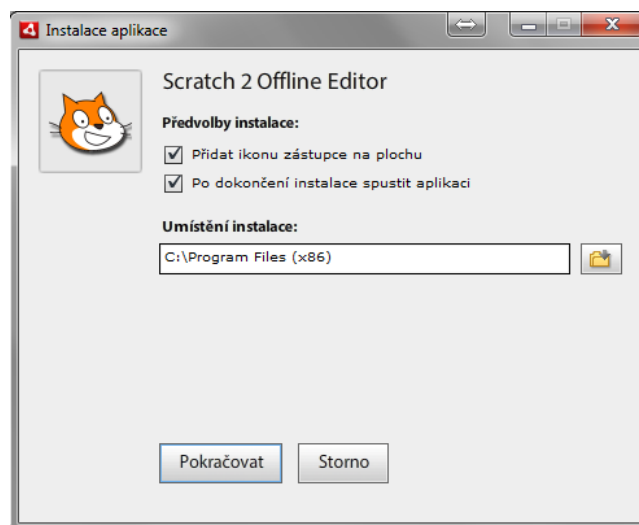
Prvním krokem při instalaci offline editoru je otevření webového prohlížeče a načtení stránky <http://scratch.mit.edu/>. V dolní části se nachází v sekci *Podpora* odkaz *Offline editor*. Po kliknutí na něj je zájemce o offline editor přesměrován na stránku, kde najde odkazy na vše potřebné k instalaci prostředí. Jmenovitě jsou to: multiplatformní prostředí *Adobe AIR*, instalační soubor pro Scratch 2.0 offline editor a jako poslední jsou volitelné materiály pro začátečníky, viz Obrázek 4.



Obrázek 4: Webová stránka s odkazy ke stažení [15]

Pro instalaci *Adobe AIR* stačí kliknout na odkaz *Adobe AIR* nebo na odkaz *Download* vedle příslušné platformy. Oba odkazy přeměrují uživatele na stejnou stránku. V pravé dolní části se nachází odkaz *Download*. Po kliknutí je stažen instalační soubor multiplatformního prostředí, který stačí pouze spustit. Instalace je velmi jednoduchá a doprovází při ní průvodce instalací.

Pro stažení instalačního souboru samotného offline editoru je potřeba, stejně jako při instalaci *Adobe AIR*, kliknout na odkaz *Download* vedle příslušné platformy. Na disk počítače je opět stažen instalační soubor, který stačí opět pouze spustit. Po spuštění se objeví dialogové okno, ve kterém si lze zvolit umístění, kam se Scratch 2.0 offline editor nainstaluje.



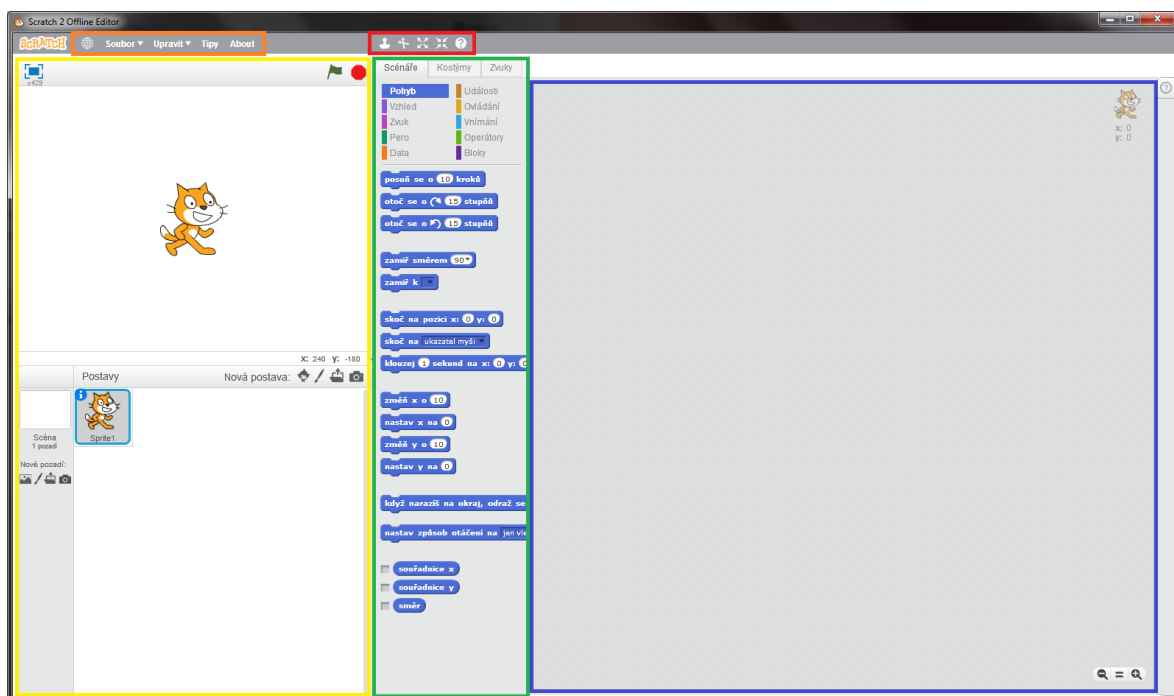
Obrázek 5: Instalace Scratch 2.0

Po výběru stačí potvrdit tlačítkem *OK*, *Scratch 2.0 offline editor* je nainstalován v řádu několika vteřin a po dokončení instalace je Scratch 2.0 offline editor ihned spuštěn. Důkladnému popisu prostředí se bude věnovat následující kapitola [15].

1.3.4 POPIS PROSTŘEDÍ SCRATCH 2.0

Při prvním spuštění právě nainstalovaného programovacího prostředí Scratch 2.0 se uživateli zobrazí okno celého prostředí, defaultně s anglickými popisky, které lze na české změnit velmi snadno kliknutím na symbol globusu vedle nápisu *Scratch* v levém horním rohu okna.

Celé prostředí je pohodlně a systematicky uspořádáno, nečiní také potíže se v něm rychle zorientovat popř. cokoli rychle najít nebo nastavit. Na obrázku 6 je vidět celé prostředí editoru Scratch 2.0, na kterém jsou vyznačeny hlavní části prostředí.



Obrázek 6: Prostředí *Scratch 2.0*

1.3.4.1 HLAVNÍ MENU

Část, která je v obrázku zvýrazněna oranžovou barvou označuje horní menu s již zmíněným symbolem globusu, který je určen pro změnu lokalizace. Dalšími položkami menu jsou: *Soubor, Upravit, Tipy a O aplikaci*.

Položka *Soubor*, pod sebou skrývá klasické operace se soubory jako je *Nový* (vytvoření nového souboru), *Otevři* (otevření dříve uloženého souboru), *Uložit* (uložení stávajícího), *Ulož jako*, *Sdílej na web* (umožňuje sdílet vytvořené projekty na webové stránce Scratche, pod jménem uživatele. Po kliknutí se zobrazí dialogové okno, které po uživateli vyžaduje *Project name* tj. jméno projektu a přihlašovací údaje ke *Scratch účtu*), *Zkontroluj aktualizace* (název položky vypovídá o funkci, která kontroluje aktuálnost nainstalované verze prostředí, popř. nabízí aktualizaci na vyšší verzi), poslední položkou je *Quit*, který zavře okno programovacího prostředí.

Položka *Upravit* pod sebou skrývá tři položky, kterými jsou *Obnovit*, *Formát malé scény* a *Turbo režim*. *Obnovit* lze využít například při nechtěném smazání postavy – spritu ze scény. Položka *Formát malé scény* zmenší scénu a podokno výběru postav, které je v obrázku 6 vyznačeno žlutě, toto je výhodné například pokud se na skriptovací ploše nachází rozsáhlý skript a je třeba se v něm zorientovat nebo něco najít bez toho, aby se uživatel posouval po skriptovací ploše pomocí horizontálního posuvníku. *Turbo režim* je položka, která dovoluje spustit projekt ve zrychleném režimu, což lze využít například při řešení rozsáhlých matematických operací nebo pro plynulejší vykreslování 3D grafiky. Turbo režim lze mimo jiné také povolit kliknutím na zelenou vlajku, zatímco je stlačena klávesa Shift. [16]

Třetí položkou hlavního menu je položka s názvem *Tipy*. Po kliknutí na tuto položku se po pravé straně okna na skriptovací ploše objeví další podokno, které v sobě skrývá nápovědu v anglickém jazyce pro základní ovládání programu. Podokno je rozděleno na tři části: *Step-by-Step Guides*, kde se lze krok za krokem na vzorových příkladech seznámit se základními operacemi a úkony v programovacím prostředí. Druhou sekcí je *How To*, kde se v rozbalovacím menu mezi položkami *Effects, Animation, Games, Stories, Music* nacházejí další návody jak v prostředí pracovat a vytvářet programy.

Poslední částí podokna je sekce *Blocks*, kde jsou podrobně rozebrány funkce všech bloků doplněné o vzorové příklady.

O aplikaci je poslední položkou hlavního menu a po kliknutí se na ni se spustí internetový prohlížeč se stránkou <https://scratch.mit.edu/about/>, na níž jsou základní informace o Scratchi. Zde opět jako v nápovědě, není česká lokalizace úplná, některé odstavce jsou napsány v českém jazyce, jiné naopak v anglickém jazyce.

1.3.4.2 PALETA NÁSTROJŮ

Další částí, která se nachází v horní části okna je část s nástroji a v obrázku 6 je označena červenou barvou. Detail je zobrazen na obrázku 7. Prvním nástrojem v řadě se symbolem razítka je nástroj pro kopírování. Kopírovat lze jak Sprity (postavy), tak posloupnosti příkazů, které jsou na skriptovací ploše. Kopírování se provádí jednoduše a to pouhým kliknutím na nástroj razítka a následně buď na konkrétní Sprite nebo na posloupnost příkazů na skriptovací ploše.

Vpravo od nástroje pro kopírování je symbol nůžek, který slouží pro vystřižení, resp. smazání. Funkčnost je obdobná jako u nástroje pro kopírování v tom smyslu, že po kliknutí



Obrázek 7: Paleta nástrojů

na symbol nůžek a následným kliknutím na Sprite nebo na posloupnost příkazů je Sprite resp. posloupnost odstraněna. Tento krok smazání je vratný pomocí volby *Obnovit*, která se nachází pod položkou menu *Upravit*.

Třetím nástrojem na paletě je zvětšení, které je reprezentováno čtyřmi šipkami směřujícími od středu směrem ven. Tento nástroj lze použít pro zvětšení velikosti Spritů na Scéně, po výběru nástroje lze na Sprite opakovaně klikat a zvětšovat tak jeho velikost až do jisté maximální velikosti.

Předposledním nástrojem je zmenšení, které je stejně jako zvětšení reprezentováno symbolem čtyř šipek, ale na rozdíl od zvětšení šipky směřují směrem dovnitř. Velikost lze zmenšovat opět po výběru nástroje opakovaným klikáním na Sprite. Obdobně jako u zvětšování je zde určitý limit, po jehož dosažení už Sprite nelze dále zmenšovat.

Šedý otazník v bílém kruhu je posledním symbolem na paletě a jedná se o nápovědu. Po kliknutí na tento symbol se kurzor změní na šedý kruh s bílým otazníkem uvnitř a po kliknutí

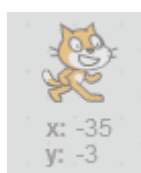
např. na některého ze Spritů v seznamu spritů vyjede z pravé strany skriptovací plochy nápověda s popisky okna editoru. Stejně jako tomu bylo u hlavního menu, ani zde není česká lokalizace úplná. Popisky okna a obrázků jsou pouze v anglickém jazyce.

1.3.4.3 SKRIPTOVACÍ OBLAST

Podokno, které zabírá největší část celého programovacího prostředí, vyznačeno modrou barvou, se nazývá *skriptovací oblast* v některých publikacích také *oblast pro scénáře*.

Jedná se o velkou plochu, na kterou jsou přesouvány a následně spojovány bloky příkazů, kterými se definuje chování Spritu nebo scény. Každý Sprite resp. scéna má svoji vlastní skriptovací oblast. Pro přesun bloku z palety bloků je třeba na blok kliknout a pomocí myši ho přetáhnout na skriptovací plochu.

Skriptovací oblast obsahuje i některé funkce, které mohou být velmi nápomocné při tvorbě scénářů pro Sprity resp. scény. Jednou z nich je zobrazení souřadnic v kartézské soustavě vybraného Spritu. Tyto souřadnice jsou k nalezení v pravém horním rohu skriptovací oblasti pod malým obrázkem, který reprezentuje aktuální Sprite, viz Obrázek 8.



Obrázek 8: Souřadnice Spritu

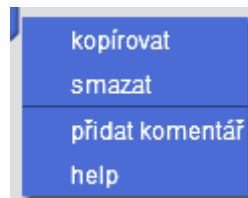
Další užitečnou funkcí je možnost oddálení či přiblížení, které všechny bloky zmenší, resp. zvětší. Ovládací prvky jsou k dispozici v pravém dolním rohu skriptovací oblasti a jsou celkem tři. Dva symboly lupy, jeden se symbolem mínusu, druhý se symbolem plusu pro oddálení resp. přiblížení. Mezi lupami je symbol rovnítka, který veškeré změny vrátí a bloky poté mají výchozí velikost. Výchozí velikost je shodná s velikostí jakou mají bloky na paletě bloků.



Obrázek 9: Ovládací prvky

Nedílnou součástí každého programovacího prostředí je možnost přidávat ke kódu komentáře, které mohou být velice nápomocné, pokud projekt otevře někdo jiný a je pro něj obtížné se v kódu zorientovat nebo ho pochopit. Další častou variantou je, kdy autor otevře projekt po delší době nečinnosti a potřebuje si rozpomenout, co který blok či úsek

kódu znamená. Možnost přidávat komentáře obsahuje samozřejmě i programovací prostředí Scratch. Komentář lze přidat kamkoli do skriptovací oblasti nebo přímo na specifický blok, který se na skriptovací ploše nachází. V obou případech postačí pravý klik myši na místo ve skriptovací oblasti, kam chceme komentář přidat nebo na příslušný blok,



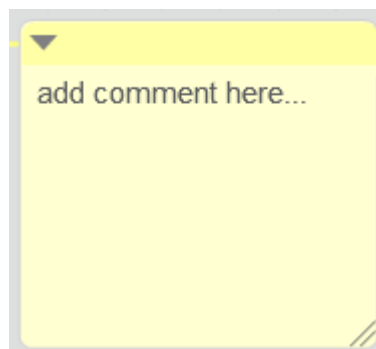
Obrázek 10: nabídka s možnostmi - blok



Obrázek 11: nabídka s možnostmi - skriptovací plocha

který chceme okomentovat.

Jak lze vidět z obrázku 10 a obrázku 11 nabídky s možnostmi se liší podle toho, co je cílem komentáře. V obou případech stačí však zvolit možnost *přidat komentář*, po jejím zvolení se objeví žluté okno s nápisem „add comment here...“, po kliknutí do okna lze přidat komentář, jak je znázorněno na obrázku 12.



Obrázek 12: okno komentáře

Pokud je komentář navázán na blok, tak se při přesouvání bloku přesouvá komentář spolu s ním. Komentář lze jednoduše smazat pravým klikem na komentář a v nabídce možností zvolit možnost *smazat*.

Další vtipnou funkcí skriptovací plochy je tzv. *Úklid*, jeho funkce spočívá v přehledném zarovnání všech bloků nacházejících se na skriptovací ploše do levého

horního rohu tak, aby se jednotlivé bloky nepřekrývaly a byly viditelné. Funkce *Úklid* je dostupná po pravém kliknutí na kterékoli prázdné místo skriptovací plochy viz Obrázek 11: nabídka s možnostmi – skriptovací plocha.

1.3.4.4 PALETA BLOKŮ

Zelená barva viz Obrázek 6: Prostředí *Scratch 2.0*, vytyčuje oblast zvanou *Paleta bloků*. V její horní části se nacházejí tři záložky: *Scénáře*, *Kostýmy* a *Zvuky*.

1.3.4.4.1 SCÉNÁŘE

Záložka scénáře v horní části obsahuje seznam kategorií, do kterých jsou rozděleny bloky podle funkce a možnosti využití. Kategorií je celkem deset: *Pohyb*, *Vzhled*, *Zvuk*, *Pero*, *Data*, *Události*, *Ovládání*, *Vnímání*, *Operátory* a *Bloky*. Kategorie jsou stejné při definici chování spritu i scény s tím rozdílem, že kategorie *Pohyb* neobsahuje pro scénu žádné bloky a v některých dalších kategoriích je počet dostupných bloků pro scénu omezen.

Kategorie *Pohyb* jak už bylo zmíněno v předcházejícím odstavci je dostupná pouze při definici chování pro sprity. V této kategorii se nacházejí bloky, pomocí nichž lze sprite rozpohybovat nejrůznějšími způsoby. Spritem lze pohybovat směrem, který má aktuálně nastavený, přičemž směr spritu lze taktéž měnit. Další možností je přidělení souřadnic, na které se sprite přesune. Tento způsob není výhodný, pokud je třeba docílit plynulého pohybu. Problém eliminuje další blok, jehož pomocí lze rovněž nadefinovat souřadnice spritu, ale je doplněn ještě o jeden parametr, který určuje, jak dlouho se bude sprite na zadané souřadnice přesouvat. Dalšími bloky lze například definovat pouze jednu ze souřadnic nebo je měnit o určitou zadanou hodnotu. Další možností je například přesun spritu na ukazatel myši popř. jiný sprite nebo nastavení směru vzhledem k jinému spritu. Kromě bloků obsahuje kategorie *Pohyb* i tři proměnné: *souřadnice x*, *souřadnice y* a *směr*.

Vzhled je první kategorií, která obsahuje bloky pro sprity i pro scénu. Nejprve se budu věnovat méně početné části, kterou je část s bloky pro scénu. Obsahuje šest bloků, kterými lze ovlivnit změnu pozadí, její barvu popř. všechny změny zrušit. Část je doplněna o dvě proměnné: *název pozadí* a *číslo pozadí*. V početnější skupině bloků pro sprite, lze měnit kostýmy spritu jeho barvu, velikost, posouvat ho více do popředí či pozadí, úplně ho skrýt a znovu zobrazit. V neposlední řadě lze také sprite rozprávět nebo zobrazit co si „myslí“,

v obou případech lze myšlenku či komentář spritu zobrazit na určitý časový úsek nebo dokud skript není ukončen.

Třetí v pořadí je *Zvuk*, ve kterém jsou bloky pro scénu a pro sprite totožné. Jak už název napovídá, s pomocí bloků z této kategorie lze do skriptů zakomponovat různé zvuky z bohaté knihovny, vlastní nahrávky, zvuk jednotlivých kláves klavíru nebo různých druhů hudebních nástrojů. Lze definovat hlasitost, dobu reprodukce, tempo nebo pauzu. Bloky jsou doplněny o dvě proměnné: *hlasitost a tempo*.

Pero je další kategorií kde se počet dostupných bloků pro scénu a sprity se liší, pokud začnu u první zmíněné scény, je zde pouze jeden blok, který smaže všechny dosud nakreslené čáry vykreslené pomocí ostatních příkazů z této kategorie. Část určená spritům je na bloky o něco štedřejší. Obsahuje bloky, s jejichž pomocí lze na scénu kreslit čáry různé tloušťky a barvy. Tato kategorie neobsahuje žádné proměnné.

Kategorie *Data*, neobsahuje žádné bloky ani proměnné, ale dvě tlačítka pro vytváření proměnných a seznamu. Po kliknutí na *Vytvořit proměnnou*, se objeví dialogové okno, ve kterém je po uživateli vyžadováno zadání názvu proměnné a určení, zda je proměnná pouze pro určitou postavu nebo pro všechny, tuto vlastnost lze vnímat jako definici lokálních a globálních proměnných. Druhé tlačítko slouží pro vytvoření seznamu, po kliknutí se opět objeví dialogové okno jako při vytváření vlastní proměnné, kde je opět vyžadováno zadání názvu a určení parametru, pro koho je seznam určen. Seznam lze vnímat jako datový typ záznam, nehraje tedy roli, jakého datového typu proměnné v něm uložené nabývají hodnotu. Pokud je vytvořena nějaká proměnná nebo seznam, zpřístupní se v kategorii další bloky, které umožní s proměnou resp. se seznamem pracovat. V případě proměnné lze proměnou nastavovat na určitou hodnotu nebo ji o určitou hodnotu změnit, proměnnou ukázat nebo skrýt či s ní dále pracovat ve skriptu. Zviditelnění či schování lze taktéž zajistit pomocí zaškrtačacího políčka vedle názvu proměnné. V případě seznamu lze prvky přidávat na konec seznamu, vkládat je na určitou pozici, nahradit prvek na určité pozici jiným prvkem nebo prvek z pozice smazat, dále jsou k dispozici dvě proměnné. Jedna z nich obsahuje hodnotu prvku na zadané pozici a druhá obsahuje délku celého seznamu. Následuje jeden logický výraz, jehož výsledkem je logická hodnota, určena přítomností či nepřítomností zadané položky v seznamu. Stejně jako u klasické proměnné lze stejným způsobem seznam skrýt či zobrazit.

Události obsahují bloky, které spouštějí skripty na základě nějaké události. První v řadě je blok, který reaguje na kliknutí na zelenou vlajku v horní části scény, dalším je blok reagující na stisknutí klávesy (směrové šipky, mezerník, písmena a čísla na numerické klávesnici), blok reagující na kliknutí na sprite, blok reagující na změnu pozadí, úroveň hlasitosti nebo obdržení zprávy. Lze definovat vlastní zprávy a následně je rozesílat a přijímat mezi sprity a scénami. Bloky z této kategorie jsou totožné pro sprity i pro scény.

Velmi důležitou kategorií je *Ovládání*. Obsahuje bloky reprezentující podmíněné příkazy a cykly (s pevným počtem opakování, nekonečný cyklus, cyklus s podmínkou na začátku). Dalšími bloky v kategorii lze skript pozastavit nebo úplně zastavit. *Ovládání* dále obsahuje bloky, jejichž pomocí lze sprity klonovat.

Vnímání je kategorie s vysokým počtem bloků a proměnných. Bloky dokáží vracet logické hodnoty na základě toho, jestli je stisknuta vybraná klávesa popř. myš, jestli se sprity dotýkají jiných spritů, kurzoru myši nebo nějaké barvy. Další možností je vrácení vzdálenosti od jiného spritu nebo ukazatele myši. Součástí jsou dále proměnné, které uchovávají hlasitost, jméno uživatele či časové údaje. Bloky pro scénu jsou téměř totožné s těmi pro sprity s tím rozdílem, že neobsahují bloky vracející logické hodnoty na základě kolize se spritem, kurzorem myši, či barvou.

Sekce *Operátory* obsahuje bloky, které provádí základní matematické operace (sčítání, odčítání, násobení a dělení), dalším v řadě je blok pro generování náhodných čísel ze zadaného intervalu, operátory porovnání, logické operátory (AND, OR, NOT), bloky pro spojování řetězců, vrácení písmena ze zadaného řetězce na zadaném indexu, dále bloky pro zaokrouhlení, blok pro matematickou operaci *Modulo* (zbytek po dělení) a posledním blokem je blok pro odmocninu. Tato sekce je opět naprosto stejná z pohledu spritu a scény.

Poslední sekci jsou *Bloky*. V sekci je možné vytvářet vlastní bloky, které lze s hlediska programování chápat jako vytváření procedur jak s parametrem, tak bez parametru.

1.3.4.4.2 KOSTÝMY - POZADÍ

V pořadí druhou záložkou jsou kostýmy resp. pozadí, bavíme-li se o spritech resp. scénách. V obou případech můžeme vybírat ze široké knihovny již předpřipravených kostýmů a pozadí, které můžeme pomocí vestavěného vektorového resp. bitmapového editoru dále upravit nebo naopak vytvořit zcela nový, vlastní kostým resp. pozadí. Dále je zde možnost

kostýmy resp. pozadí nahrát ze souboru nebo pořídit nový obrázek pomocí webkamery počítače.

1.3.4.4.3 ZVUKY

Aby byly skripty o něco veselejší a zábavnější, obsahuje *Scratch* také možnost přidávání zvuků. Opět se zde nachází rozsáhlá knihovna defaultních zvuků, pokud by si uživatel nevystačil s předem připravenými zvuky, může využít možnost nahrání vlastního záznamu z mikrofону, který lze posléze doplnit o nejrůznější efekty. Nechybí zde ani možnost výběru souboru se zvukem. Zvuky jsou dostupné jak pro sprity, tak pro scénu. Nutno zmínit, že k pořízeným nahrávkám či přihraným souborům lze poté přistupovat a zakomponovat je do bloků sekce *Zvuk*.

1.3.4.5 SCÉNA, SEZNAM POSTAV

1.3.4.5.1 SCÉNA

Poslední oblastí, kterou Obrázek 6: Prostředí *Scratch 2.0* zobrazuje a je vyznačena modrou barvou, je část pro scénu a seznam postav. Scéna se nachází v horní polovině vyznačené oblasti a je to místo, kde se sprity pohybují, chovají se podle skriptů nadefinovaných na *Skriptovací ploše* a reagují na ostatní sprity.

V horní části scény na pravé straně je symbol zelené vlajky, který slouží pro spuštění skriptu, viz blok z kategorie *Události*. Vedle něj se nachází symbol červeného osmiúhelníku, který skript zastavuje. Naproti symbolům vlajky a osmiúhelníku se na levé straně nachází symbol obdélníku s přerušovaným obrysem, po kliknutí na něj je scéna zvětšena přes celou obrazovku. Této vlastnosti lze využít při např. při prezentacích.

Pod pravým dolním rohem scény se nacházejí souřadnice, které nerepresentují souřadnice spritu, jako tomu bylo u *Skriptovací plochy*, ale popisují aktuální pozici kurzoru myši na scéně.

1.3.4.5.2 SEZNAM POSTAV

Nachází se v levém dolním rohu celého prostředí a je zde seznam všech spritů, které jsou v aktuálním projektu. Sprity lze přidávat pomocí menu, které je vedle popisku „*Nová postava:*“. Sprity lze podobně jako zvuky či kostýmy vybírat z bohaté knihovny nakreslit

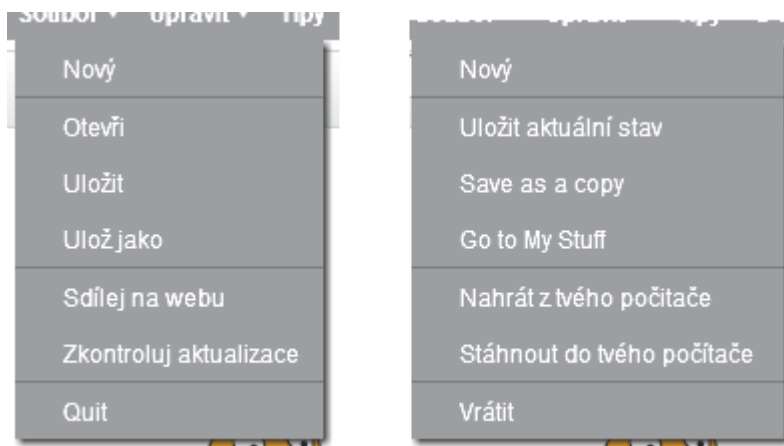
pomocí vektorového či bitmapového editoru vlastní, nahrát ze souboru, či pořídit snímek webovou kamerou a pořízený snímek použít pro nový sprite.

Po kliknutí na sprite v seznamu se sprite zvýrazní modrým rámečkem a na *Skriptovací ploše* se zobrazí skripty, kterými má sprity nadefinované chování. Zároveň se v horním levém rohu rámečku objeví symbol „i“ v modrém kruhu. Po kliknutí na něj se seznam postav schová a je nahrazen menu, kde má uživatel možnost změnit jméno spritu, styl otáčení, viditelnost spritu na skriptovací ploše a vlastnost zda může být sprite přesouván hráčem. Zároveň jsou zde informativně zobrazeny aktuální souřadnice spritu a jeho směr. Zpět na seznam postav se lze vrátit kliknutím na symbol *zpět* v levém horním rohu.

Vlevo od seznamu postav je část věnovaná úpravě scény, přidávání a modifikaci jednotlivých pozadí. Podobně jako u postav, lze pozadí vybírat z knihovny, vytvořit vlastní pomocí vestavěného editoru, vložit ze souboru či pořídit snímek pomocí webkamery.

1.3.5 POROVNÁNÍ ONLINE A OFFLINE EDITORU

Oba dva editory jsou v zásadě stejně avšak s několika rozdíly, hned prvním jak je patrné z obrázku 13. Dalšími rozdíly jsou: absence *Batohu* u offline verze a nemožnost vytváření cloudových proměnných. Vytváření cloudových proměnných je dále podmíněno jistým statusem účtu na webu Scratche. Každý nový uživatel začíná s hodnotí *New Scratcher*, pro vytváření cloudových proměnných je mít hodnost *Scratcher* [14].



Obrázek 13: Porovnání položek menu *Soubor*

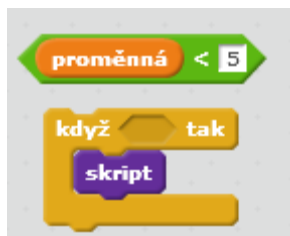
1.4 MOŽNOSTI VYUŽITÍ PROGRAMU

Scratch je se svojí jednoduchostí, pěkným uživatelským rozhraním a animovanými postavami předurčen k výuce programování pro úplné začátečníky nebo děti na základních školách.

1.4.1 VÝUKA ZÁKLADŮ PROGRAMOVÁNÍ

Pokaždé když se člověk učí nějaké nové dovednosti, není lepší způsob než postupovat postupně po malých krocích od jednoduchých, snadno pochopitelných věcí až k těm složitým a více komplexním. Není tomu jinak ani při výuce programování. Začít s programováním za pomoci jazyka Scratch se jeví jako velmi výhodné. Rozpohybování animované postavy přesunem několika málo bloků na skriptovací plochu je velmi jednoduché. Dalším, v počátcích velmi důležitým a velmi motivujícím důvodem pro pokračování je fakt, že výsledky snažení jsou velmi rychle vidět. Začátečník je pak motivován k dalšímu učení a hlubšímu seznamování se s programováním. Odpadá jakákoli nutnost znalosti syntaxe, což je pro všechny začátečníky nesporná výhoda. Začátečník se pak může soustředit na jádro problému a neztrácí čas a motivaci hledáním a opravováním

syntaktických chyb. Jednotlivé bloky, které je možné mezi sebou kombinovat, do sebe zapadají podobným způsobem jako puzzle. Pro představu blok reprezentující podmínku *if* v sobě má tvar šestiúhelníku, do kterého „zapadne“ pouze blok stejného tvaru, který reprezentuje logický výraz, viz obrázek 14.



Obrázek 14: Demonstrace tvaru bloků

Pod pojmem základy programování si představuji pochopení základních principů, ke kterým patří jednoduché příkazy, datové typy, podmínky a cykly.

Scratch může posloužit nejenom k výuce programování, ale i k rozšíření ostatních dovedností. Mezi tyto dovednosti se podle mého názoru řadí kreativita, schopnost kooperace, základy algoritmizace, chápání věcí v širších souvislostech nebo promítnutí principů (např. jak funguje podmínka) osvojených při tvoření ve Scratchi do dalších činností, které jsou spojené s užíváním výpočetní techniky.

V dnešní době je stále více kladen důraz na schopnost kreativity a její rozvíjení. Scratch tuto schopnost stimuluje nenásilnou formou. Žáci si tvořením jednoduchých skriptů a her tuto schopnost procvičují a dále rozvíjejí, aniž by si to uvědomovali. Proces učení zde může mít zábavnou formu a žáci jsou dále motivováni k další činnosti.

Schopnost pracovat v týmu je další dovedností, kterou lze pomocí programu Scratch rozvíjet. Práci na projektech lze rozdělit mezi žáky ve třídě. Pokud je uživatel již pokročilejší, může se zapojit do tvorby složitějších a komplexnějších projektů, na kterých pracuje komunita kolem programu. Nabídky takovýchto kooperací se nacházejí na diskuzních fórech a je jim věnováno samostatné vlákno. Vzhledem k tomu, že komunita je velmi široká a příspěvky jsou většinou v anglickém jazyce, může být dalším benefitem i procvičení cizích jazyků.

1.4.2 TVORBA INTERAKTIVNÍCH VÝUKOVÝCH PREZENTACÍ

Scratch mohou efektivně také využít pedagogové, kteří přímo programování nevyučují, ale díky jednoduchosti Scratche, jsou si schopni sami velmi snadno vytvářet různé výukové materiály nebo interaktivní prezentace. Příkladem může být hra Třídění odpadu.

Prezentace nemusejí učitelé přímo sami vytvářet, lze využít nespočet již vytvořených prezentací a výukových her, které jsou sdílené na webu Scratche. Web jsem prozkoumal a vybral jsem si několik zdařilých projektů, které představím.

Prvním projektem, který mě zaujal je *waves*. Jedná se o velmi jednoduché vykreslování sinusoidy, které lze zároveň měnit amplitudu a frekvenci. Vykreslování reaguje dynamicky na změny obou parametrů. Projekt by mohl posloužit v některých předmětech např. elektrotechnika, kdy si žáci mohou vyzkoušet změna amplitudy a frekvence ovlivní výsledný signál. Odkaz na projekt na webu: <https://scratch.mit.edu/projects/346204/>

Dalším projektem, který funguje na podobném principu jako *waves*, je *Interference of waves – Laboratory*. Projekt vtipně demonstruje skládání dvou kmitů. Stejně jako u *waves*, lze měnit amplitudu, frekvenci a navíc je zde i fázový posun. Odkaz na projekt na webu: <https://scratch.mit.edu/projects/57764570/>

Projekt pro výuku fyziky resp. optiky se jmenuje *Lenses 1.6*, dostupný je na adrese: <https://scratch.mit.edu/projects/779119/>. Simuluje chování čoček v závislosti na zadaných parametrech (tloušťka, ohniskové vzdálenosti, mohutnost, ...).

Abych nezůstal pouze u projektů týkajících se fyziky, vybral jsem si jeden, který je z oblasti zeměpisu. Cílem je procvičení zemí Evropy. Cílem je doplnit chybějící jména zemí na základě polohy na mapě a vlajky. Odkaz: <https://scratch.mit.edu/projects/16074797/>

1.4.3 SCRATCH A OOP

Zkratka OOP značí objektivě orientované programování. Umožňuje vytváření tříd, které definují datové typy, které jsou známé pod pojmem *objekt*.

Objekty se vytvářejí podle vlastností definovaných ve *třídě*, kterou si tedy lze představit jako jakousi šablonu, podle níž je možné vytvořit libovolný počet objektů. *Instance třídy* je poté jeden konkrétní objekt, který je podle šablony vytvořený.

Mezi klíčové vlastnosti objektivě orientovaného programování patří dědičnost. Tato vlastnost umožňuje třídám (potomkům) dědit vlastnosti od nadřazené třídy (rodič). Pro příklad si lze představit třídu *Ovoce*, jejímž potomkem bude třída *Jablko*.

Polymorfismus je další klíčovou vlastností OOP. Herout polymorfismus vysvětluje následovně: „Slovo **polymorfismus** lze nejlépe vyjádřit českým slovem **vícetvarost** nebo **mnohotvarost**. Jedná se o možnost využívat v programovém textu stejnou syntaktickou podobu pro metody s různou vnitřní reprezentací – jinak řečeno, voláme jakoby pořád stejnou metodu, ale ta provádí pokaždé něco jiného. Polymorfismus dovoluje jednotným způsobem manipulovat s prvky příbuzného, ale předem neznámého typu. Teorie ohledně polymorfismu je značně propracovaná ...“ (HEROUT, Pavel. *Učebnice jazyka Java*. 5., rozš. vyd. České Budějovice: Kopp, 2010, 386 s. ISBN 978-80-7232-398-2.)

Třetí klíčovou vlastností je podle Herouta: „**Zapouzdření dat** (data encapsulation) je jedním z tří vždy zmiňovaných pilířů objektivě orientovaných jazyků. Jedná se o explicitní spojení dat a metod dané konstrukcí třídy, tzn. data se vždy vyskytují společně s metodami, které s nimi manipulují.“ (HEROUT, Pavel. *Učebnice jazyka Java*. 5., rozš. vyd. České Budějovice: Kopp, 2010, 386 s. ISBN 978-80-7232-398-2.)

Názory na to, jestli je programovací jazyk Scratch objektivě orientovaný, či nikoli se liší. Některé jeho vlastnosti ho řadí mezi objektivě orientované jazyky, některé další se ale s principy OOP rozcházejí.

Sprity (postavy) použité ve Scratchi lze jednoznačně vnímat jako použití objektů jako je tomu u OOP. K vlastnostem postav lze přistupovat prostřednictvím ostatních postav. Od verze *Scratch 2.0* je k dispozici klonování spritů. Když už jsem zmínil klonování spritů, dostávám se k jednomu velkému nedostatku a to je přistupování k dynamicky vytvořeným klonům. Přístup k takto vytvořeným klonům je možný hned po jejich vytvoření.

Opět se zmíním o věkové kategorii, která tvoří nejpočetnější část uživatelů, a to jsou začátečníci ve věku 10 až 15 let. Pro uživatele, kteří se s programováním seznamují a Scratch je jejich prvním programovacím jazykem je podle mého názoru zvolená koncepce postav jakožto objektů vhodná. Lépe si pak dokáží spojit všechny souvislosti, když se po zvládnutí úvodu do programování v jazyce Scratch přesunou k plnohodnotnému objektově orientovanému jazyku.

2 UŽIVATELSKÁ KOMUNITA

2.1 SCRATCH ÚČET

Před začátkem samotného programování ve Scratchi všem doporučuji vytvořit si vlastní Scratch účet. Pokud plánujeme využití Online editoru, je Scratch účet velmi dobrý nástroj pro ukládání a správu našich vytvořených projektů. Projekty lze ukládat na server a je k nim také možné přistupovat téměř z jakéhokoli počítače s připojením na internet. Pro tuto možnost se nabízí přímé uplatnění ve výuce. Žáci, kteří by pracovali ve Scratchi ve škole, by pak nemuseli přenášet žádné soubory ze školy domů a naopak, například při plnění domácích úkolů nebo jiných prací.

Vytvořené projekty lze i sdílet s ostatními uživateli programu. Můžete tak prohlížet některá povedená díla ostatních uživatelů a v mnohém se přiučit nebo inspirovat pro svoji další tvorbu.

2.1.1 VYTVOŘENÍ SCRATCH ÚČTU

Vytvoření účtu je velmi jednoduché, postup je následující: v internetovém prohlížeči přejděte na stránku <http://scratch.mit.edu/>, v pravém horním rohu je odkaz *Připojit se ke Scratchi*, po kliknutí se zobrazí dialogové okno, ve kterém jste vyzváni k zadání uživatelského jména, prostřednictvím kterého se budete přihlašovat ke svému účtu a v neposlední řadě také heslo a jeho potvrzení. Po vyplnění těchto položek a kliknutí na tlačítko další je uživatel vyzván k zadání dalších údajů, konkrétně jsou to: datum narození, pohlaví a země, ze které uživatel pochází. Dalším krokem je už pouze vyplnění emailové adresy, na kterou je zaslán potvrzovací email. V této fázi je účet hotový, uživatel je automaticky přihlášen a může začít s programováním [17].

2.2 SCRATCH WIKI

První verze Scratch Wiki byla vytvořena 6. prosince 2008. Tuto Wiki vytvořil jeden z uživatelů programovacího jazyka Scratch. Wiki vznikla bez vědomí a asistence vývojářů Scratche. Vývojáři si této Wiki nakonec všimli a chtěli ji umístit na své webové stránky.

Wiki je dostupná zdarma a obsahuje nejrůznější informace o programování ve Scratchi, jeho historii. Wiki je vývojáři podporována, ale mezi přispívatele patří z větší části uživatelé

Scratche. Aby se uživatel Scratche mohl účastnit přispívání na Wiki, musí si zažádat o speciální účet k tomu určený.

Wiki je v anglickém jazyce a vývojáři podporují uživatele, jejichž mateřským jazykem není angličtina, aby vytvořili vlastní Wiki ve svém rodném jazyce. Prvním takovýmto cizojazyčným projektem byla Wiki pro německy mluvící uživatele z Německa, Rakouska a Švýcarska, která byla vytvořena v únoru roku 2012. Mezi další cizojazyčné projekty patří Wiki pro uživatele z Indonésie, Ruska a Japonska [18] [19].

2.3 DISKUZNÍ FÓRA

Scratch pro svoji početnou komunitu příznivců z řad žáků i učitelů zřídil velmi rozsáhlé fórum, které lze nalézt na adrese <http://scratch.mit.edu/discuss/> nebo po kliknutí na odkaz *Diskutovat* na hlavní stránce Scratche. Fórum je rozděleno do několika základních částí, které budou v několika následujících podkapitolách představeny a popsány. Samozřejmostí je možnost přihlášení se k RSS kanálu a mít tak přehled o nejnovějších příspěvcích a možnost zobrazení celého fóra v režimu pro mobilní zařízení.

2.3.1 WELCOME TO SCRATCH

První část diskuzního fóra je rozčleněna na dvě vlákna. Prvním vláknem jsou *Announcements*, kde jsou uvedeny nejrůznější aktuality a updaty týkající se Scratche a také některé hotfixy, které právě vyšly. Toto vlákno je velmi často aktualizováno.

Další vlákno, jak už jeho samotný název napovídá, *New Scratchers*, je určeno všem novým lidem, kteří se rozhodli připojit se ke komunitě. Jsou zde objasněny často kladené dotazy nejrůznějšího charakteru, včetně návodů co, kde a jak nastavit. Odezva od ostatních uživatelů je v případě kladení nějakého dotazu si troufám tvrdit velmi rychlá.

2.3.2 MAKING SCRATCH PROJECTS

Tato část fóra je věnována vzájemné pomoci a spolupráci na projektech tvořených v programu Scratch, v jednotlivých vláknech si lze nechat poradit s nefunkčním skriptem či projektem nebo se naopak pochlubit svými výtvořky a podiskutovat o nich s ostatními uživateli. Jedno vlákno je věnováno některým nápadům, které by se daly pomocí Scratche realizovat. Tyto idey lze nabídnout ostatním nebo si naopak nechat od nějakého kreativnějšího Scratchera poradit do čeho zajímavého se dál pustit.

Pokud má uživatel chuť a ambice pustit se do něčeho složitějšího, ale nechce na projektu pracovat sám je pro něj určeno vlákno *Collaboration*, ve kterém někteří další uživatelé hledají další kolegy, se kterými by v týmu mohli pracovat na často náročnějších a ve většině případů velmi zajímavých projektech.

Posledním vláknem této části je vlákno s názvem *Requests*. V tomto vlákně další uživatelé nabízejí nejrůznější služby, např. vytváření projektů na zakázku, tvorbu grafických podkladů pro projekty a mnoho jiného.

2.3.3 ABOUT SCRATCH

Část je věnována základním otázkám, které se týkají programu Scratch. Nachází se zde celkem sedm vláken. Prvním z nich jsou *Questions about Scratch*, v tomto vlákně uživatelé diskutují nad základními otázkami a problémy, spíše technického rázu. Vlákno je velmi rozsáhlé a odezva od ostatních uživatelů je opravdu velmi rychlá. *Suggestions*, je další vlákno, jenž je určeno pro podávání návrhů na zlepšení či doplnění programu z řad uživatelů. Třetím vláknem jsou *Bugs and Glitches*, ze samotného názvu už lze snadno odvodit, že vlákno je určeno pro reportování nejrůznějších chyb, se kterými se uživatelé setkávají během svého tvoření.

2.3.4 INTERESTS BEYOND SCRATCH

Předposlední část je určena k diskuzi o věcech, které se Scratchem nemají žádnou spojitost. *Things I'm Making and Creating*, je vlákno, kde se uživatelé diskutují nad ostatními věcmi mimo Scratch. Velká část příspěvků se týká jiných programovacích jazyků. Dalšími probíranými tématy jsou například hudba nebo různé koníčky. Druhým a zároveň také posledním vláknem této kategorie je vlákno nesoucí název *Things I'm Reading and Playing*. Příspěvky jsou velmi různorodé, ale převládají ty, ve kterých jsou zmíněné počítačové, ale i jiné hry, kterými se uživatelé baví ve volném čase.

2.3.5 SCRATCH IN OTHER LANGUAGES

Poslední část fóra věnované příspěvkům v jiných jazycích než je angličtina, skladba příspěvků je velmi různorodá. Nechybí zde ani příspěvky od českých uživatelů.

2.4 SCRATCHED

Své místo v komunitě uživatelů mají i učitelé a lidé, kteří Scratch vyučují. Pro ně je určena stránka *ScratchEd*, Scratch for Educators. Před vznikem této stránky si učitelé a lektoři jazyka vyměňovali své zkušenosti a nápady na oficiálních fórech, kde měli své vlákno *Educators*. Se stále se zvětšujícím počtem uživatelů si tým stojící za vývojem Scratche uvědomil, že bude lepší vytvořit samostatný prostor, kde si učitelé a lektoři budou moci vyměňovat své zkušenosti, materiály, další zdroje a kde budou moci diskutovat. Vznikla tedy samostatná webová stránka <http://scratched.gse.harvard.edu/>. Stránku lze rozdělit na tři hlavní části: *Stories*, kde uživatelé sdílí své osobní zkušenosti s výukou Scratche. *Resources* je část, kde se nacházejí materiály, které učitelé mohou využít. Jsou rozdělené do několika kategorií podle stupně vzdělání, typu obsahu, druhu oboru nebo podle jazyka. Poslední částí je samostatné fórum pro učitele a lektory. Pro přispívání na ScratchEd je potřeba mít vytvořený ScratchEd účet, který slouží pouze pro ScratchEd. Pokud jde o dostupnost výše zmíněných materiálů, není nutné mít vytvořený ScratchEd účet, materiály jsou dostupné volně [20] [21].

2.5 SCRATCHJR

ScratchJR je aplikace pro tablety s operačním systémem Android a iOS. Byla uvedena 30. července 2014 a na jejím vývoji se podílela MIT a *Tufts University*. Aplikace přejímá principy a myšlenky programovacího jazyka Scratch. Stejně jako u Scratche jsou zde postavy, kterým je možné vdechnout život za pomoci skládání příkazů ve formě bloků. Grafická stránka, vzhled a funkce bloků a postav jsou optimalizovány, aby zaujaly oko mladšího uživatele (5-7 let). Děti si tímto způsobem už od útlého věku mohou programovat vlastní hry a jednoduché skripty, čímž si rozvíjejí kompetence pro utřídění myšlenek, kreativitu a logické myšlení. Nenásilnou formou hry sami sebe uvádějí do problematiky algoritmizace, matematiky a řešení problémů. Vytvoření ještě jednoduššího a přístupnějšího programovacího prostředí pro mladší uživatele bylo podle mě dobrým krokem. Mladší uživatelé mají takto možnost osvojit si práci s výpočetní technikou a principy jejího fungování už v útlém věku a je pro ně pak jednodušší přejít na složitější programovací jazyky a dobře je ovládnout. Myslím, že já i moji kolegové bychom s programováním byli na vyšší úrovni, kdybychom měli možnost začít už v takhle útlém věku pomocí něčeho jako je ScratchJR. Dalším dobrým krokem je zacílení na tablety, které jsou

děti vnímány jako prostředek k zábavě resp. hraní. Jako hraní pak berou i tvoření v aplikaci ScratchJR [22] Obrázek prostředí je vidět na obrázku 15.



Obrázek 15: Prostředí ScratchJR [23]

3 MOŽNOSTI VYUŽITÍ – UKÁZKA PRAKTICKÝCH PŘÍKLADŮ

Tato kapitola je věnována ukázkám praktických příkladů. U každého příkladu je definováno zadání, cíle příkladu a možný návrh pro jeho řešení.

3.1 PRVNÍ PROJEKT

První projekt, sloužící k seznámení s programovacím prostředím a několika bloky.

3.1.1 CÍLE

V tomto úvodním projektu si žáci osvojí práci s několika základními bloky. Poprvé rozpohybují sprite (postavu) a naučí se vykreslit dráhu jejího pohybu.

3.1.2 POPIS ŘEŠENÍ

Vzhledem k tomu, že se jedná o prvotní projekt, ve kterém se žáci budou teprve pomalu seznamovat s programovacím prostředím Scratch, obsahuje pouze jednu postavu. Postavě je definován pohyb, jehož dráha je zároveň po jednotlivých krocích vykreslována na scénu, aby si žáci udělali lepší představu, jak přesně se postava po scéně pohybuje.

3.2 TŘÍDĚNÍ ODPADU

Jednoduchá výuková hra, jejímž cílem je správně roztrždit odpad do příslušných kontejnerů. Hra tohoto typu může sloužit pro učitele přírodopisu nebo ekologie na prvním stupni základní školy, kde se žáci seznamují s tříděním odpadu. Pomocí této hry si žáci mohou procvičit nabyté znalosti. Další možností může být zapojení starších žáků, kteří by tuto edukativní hru vytvořili pro své mladší spolužáky.

3.2.1 CÍLE

Cílem je osvojit si následující dovednosti: pozicování postav, práci s kostýmy, práci se základními cykly a podmínkami a seznámí se s realizací funkce *drag and drop*.

3.2.2 POPIS ŘEŠENÍ

Prvním krokem bylo vytvoření grafických podkladů ve vektorovém editoru Inkscape a úprava obrázků v bitmapovém editoru GIMP. Samotný program je velmi jednoduchý a základními postavami jsou postavy reprezentující kontejnery pro třídění odpadu, jejichž kostýmy jsem vytvořil v editoru Inkscape. Jako první krok jsem si určil počáteční pozice všech postav. Všechny reagují změnou kostýmu na zprávu, která informuje o správném

umístění odpadu do kontejneru. Dalšími postavami jsou postavy, které reprezentují samotný odpad. U těchto postav je nutné vyřešit posouvání pomocí kurzoru myši. Dalším bodem na seznamu je podmínka, která kontroluje, jestli se příslušný odpad dotýká kontejneru, do které patří (správné roztřídění odpadu) a následné skrytí postavy.



Obrázek 16: Třídění odpadu

3.3 BLUDIŠTĚ

Cílem této hry je projít s postavou členitým labyrintem ze začátku na konec. Při postupování bludištěm má hráč k dispozici lucernu. S postupujícím časem se velikost prostoru, který je lucernou osvětlován zmenšuje. Tomu lze zabránit opětovným dobitím, které je realizováno interakcí s určitou postavou. Úroveň nabití je signalizována další postavou. Příklad patří ke komplexnějším a je tedy určen pro pokročilejší uživatele.

3.3.1 CÍLE

Žáci si procvičí pokročilejší práci s podmínkami a cykly, interakci postav, posílání zpráv mezi postavami a reakce na obdržené zprávy, práci s kostýmy, práci s proměnnými, reakce na stisknuté klávesy, logické výrazy a práci se souřadnicemi.

Příklad je vhodný pro pokročilejší uživatele nebo by bylo vhodné ho zařadit jako projekt pro zájmový kroužek.

3.3.2 POPIS ŘEŠENÍ

Před samotnou prací v programu Scratch jsem si připravil grafické podklady pomocí vektorového grafického editoru Inkscape. V programu Inkscape jsem vytvořil kostýmy pro

postavy *baterie*, *bludiste* a *blesk*. Vektorový editor jsem zvolil kvůli tomu, že jsem s ním měl čerstvé zkušenosti z předmětu KVD/ZTGI, kde jsem poznal i jeho výhody.

Ve hře je celkem jedenáct postav (*kocour*, *cíl*, *bludiste*, *pochoden*, *baterie*, *blesk*, *blesk2*, *blesk3*, *blesk4*, *blesk5* a *blesk6*). Scéna má černé pozadí, které dokresluje spolu s černým kostýmem postavy *bludiste* iluzi, že postava *kocour* prochází bludištěm ve tmě. Postava *kocour* má nadefinované reakce na směrové klávesy tak, aby se mohla bludištěm pohybovat. Nechybí ani reakce na interakci s postavou *bludiste*, aby bylo zajištěno, že se postava bude moci pohybovat pouze v místech, kde je to dovolené. Postava *kocour* dále rozesílá zprávy ostatním postavám v případě, kdy narazí na některou z postav *blesk*. Postava *cíl* slouží k indikaci dosažení konce bludiště a reaguje na postavu *kocour*. Postava *bludiste* nemá definováno žádné chování. Postava *pochoden* resp. její kostým byl za účelem testování vytvořen ve vektorovém editoru kostýmu programu Scratch. Velikost kostýmu postavy se po časovém intervalu zmenšuje, což budí dojem že *pochoden* svítí s ubývajícím časem stále méně. Podle úrovně velikosti rozesílá postava zprávy, na které reaguje postava *baterie* změnou kostýmu. Postavy *blesk* až *blesk6*, reagují podle vzdálenosti od postavy *kocour*. Pokud vzdálenost klesne pod určitou úroveň postavy *blesk* až *blesk6* se zviditelní a je možné na ně vstoupit, tím doplnit baterii a rozjasnit světlo.



Obrázek 17: Bludiště

3.4 SKÁKÁNÍ NA PLATFORMY

Komplexní příklad, ve kterém je cílem proskákat s postavou přes platformy, které se po určitém proměnném čase zhroutí a je tedy nutné postupovat rychle. Hra je vhodná pro pokročilejší uživatele, kteří už mají základní znalosti.

3.4.1 CÍLE

Cílem je osvojit si následující dovednosti: čtení a reakce na stisknuté klávesy, cykly, podmínky, práce s kostýmy, práce s proměnnými, posílání a reakce na zprávy, práce s blokem generování náhodných čísel ze zadaného intervalu, práce se scénami, vytváření vlastních bloků a klonování.

3.4.2 POPIS ŘEŠENÍ

Na rozdíl od Bludiště, jsem v tomto příkladu využil postavy z knihovny, které jsem posléze upravil pomocí bitmapového editoru ve Scratchi.

Ve hře jsou tři postavy (*kocour*, *platforma* a *průvodce*). Prvním krokem bylo vytvoření sady tří pozadí, které reprezentují tři úrovně postupu ve hře. Pro vytvoření těchto pozadí jsem využil bitmapového editoru pozadí přímo v programu Scratch. Na všech pozadích je černou barvou znázorněna startovací plošina a červenou barvou láva, které se *kocour* snaží vyhnout, aby nepřišel o jeden ze svých životů. Postava *kocour* reaguje na stisknuté kurzorové šipky pohybem vpravo (šipka vpravo), vlevo (šipka vpravo) a výskokem (šipka nahoru). Prvním krokem bylo určení výchozí startovací pozice na startovací platformu, která je na pozadí reprezentována černou barvou. Následovalo vyřešení problému s gravitací, aby bylo možné dosáhnout efektu, že *kocour* skáče. *Kocour* má dále definovány reakce na dotek s lávou a dotek s modrým praporkem, který reprezentuje cíl. V případě, že se *kocour* dotkne lávy, je mu ubrán jeden život a úroveň je restartována. Pokud postava úspěšně přeskáče přes platformy je rozeslána zpráva, na kterou reaguje jak scéna (změnou pozadí další úrovně) tak postava *platforma*. U postavy *platforma* je využito klonování sebe sama. Klonování probíhá vždy při změně scény nebo startu nové hry popř. kola. Ihned po vytvoření klonu je klon postavy napozicován. Toto pozicování je závislé na právě použitém pozadí. *Platforma* a její klony reagují na zprávu vyslanou postavou *kocour* při jejich kontaktu. Reakce spočívá ve změně kostýmu po náhodně generovaném čase a zmizením,

čímž je docíleno efektu pomalého hroucení platformem. Poslední postavou je *průvodce*. Slouží pouze jako postava pro interakci s uživatelem a informování o průběhu hry.



Obrázek 18: Skákání na platformy

3.5 HRA STRÍLENÍ NA NETOPÝRY

Další ze série komplexních příkladů pro pokročilejší uživatele. V dolní části scény se ze strany na stranu pohybuje šíp vystřelující hlaveň, která po stisku určené klávesy vystřelí směrem vzhůru šíp. Úkolem hráče je zasáhnout vystřeleným šípem netopýra, který se pohybuje nepravidelným pohybem. Po zásahu se netopýr rozdělí na dva menší. Hra končí po zasažení všech netopýrů nebo vyčerpání dostupných šípů.

3.5.1 CÍLE

Cílem této hry je osvojit si následující dovednosti: práce s kostýmy, pozicování a souřadnice, práce s podmínkami, práce s cykly posílání a reakce na zprávy, reakce na stisknuté klávesy, tvorba vlastních bloků, práce s blokem pro generování náhodného čísla ze zadaného intervalu, klonování a základní práce s klony.

3.5.2 NÁVRH ŘEŠENÍ

Prvním krokem, stejně jako u Třídění odpadu a Bludiště bylo vytvoření grafických podkladů. Opět jsem využil vektorový editor Inkscape, ve kterém jsem vytvořil kostýmy pro postavy *hlaveň* a *šíp*. Poslední postavou, která se ve hře vyskytuje, je postava *netopýr*, jejíž kostým jsem vybral z knihovny dostupných kostýmů.

Nejprve jsem definoval chování postavě *hlaveň*, která se v dolní části obrazovky pohybuje ze strany na stranu a zároveň slouží jako postava k interakci s uživatelem. Jako další následovala postava *šíp*. Skript pro tuto postavu je definován tak, že je postava na začátku naklonována a tento klon se neustále pohybuje spolu s postavou *hlaveň*. Dále je definována reakce na klávesu mezerník, kdy při jejím stisknutí je klon postavy *šíp* vystřelen směrem vzhůru a je zároveň snížena hodnota proměnné *početŠípů*. Skript zároveň pomocí zprávy informuje ostatní postavy o ukončení hry (vyčerpání dostupných šípů). Poslední postava, *netopýr*, se po startu hry nahodile pohybuje v horní části scény proměnlivou rychlostí. U této postavy je využito klonování, aby bylo docíleno efektu, tak jak je definováno v zadání příkladu.



Obrázek 19: Střílení na netopýry

3.6 ANIMACE

Animace doplňují některé z příkladů a demonstrují postup práce v programu Scratch. Všechny animace jsou součástí příloh.

3.6.1 PRVNÍ PROJEKT

Pomocí této animace je demonstrováno řešení prvního příkladu praktické části viz První projekt. Řešení tohoto příkladu je v animaci popsáno od začátku až do konce po jednotlivých krocích. V závěru animace je zobrazen i konečný výsledek. Výsledná animace je součástí příloh.

3.6.2 PŘIDÁVÁNÍ POSTAV, POZICOVÁNÍ, INTERAKCE, POSÍLÁNÍ ZPRÁV

Animace koresponduje s příkladem Třídění odpadu a demonstruje základní kroky k jeho vytvoření. Nejprve je demonstrováno přidávání nových postav, úprava jejich velikosti, pozicování, reakce na zprávy a jejich posílání. Dále je demonstrován princip ošetření interakce postav a práce s cyklem. V závěru animace je otestována funkčnost celého skriptu.

3.6.3 TVORBA BLOKŮ

Tato animace slouží k demonstraci postupu při vytváření vlastních bloků. Postup, který animace zobrazuje, jsem použil u příkladu Skákání na platformy. Animace konkrétně zobrazuje kroky nutné pro vytvoření vlastního bloku s dvěma parametry. V tomto případě se jedná o souřadnice x a y, na kterých je umístěna platforma.

ZÁVĚR

Cílem bakalářské práce bylo představit program Scratch, popsat možnosti jeho využití a vytvořit vhodné příklady, které budou tyto možnosti demonstrovat. Práce je doplněna o animace, které demonstrují postup práce v programu Scratch.

V bakalářské práci jsem nejprve popsal co Scratch je a pro koho je určen. Následně jsem popsal historii vývoje programovacího prostředí Scratch od prvních verzí až po současnou. V další části práce bylo popsáno uživatelské rozhraní programu Scratch, které jsem rozdělil do několika částí a ty důkladně analyzoval a popsal. V další kapitole se práce věnuje možnostem využití programu. Zmínil jsem se o vztahu programování ve Scratchi s objektově orientovaným programováním.

Další kapitolu jsem věnoval uživatelské komunitě, která hraje v případě programu Scratch velkou roli a nemohl jsem ji opomenout. Zmínil jsem obsah diskuzních fór a další projekty komunity.

V poslední části práce se věnuji praktickým ukázkám, na kterých demonstruji možnosti využití programu Scratch. Pro tento účel jsem vytvořil několik příkladů. U každého příkladu jsem definoval zadání, následně co je cílem příkladu a jako poslední jsem nastínil možný návrh řešení daného příkladu.

RESUMÉ

The thesis is introducing the program Scratch, describes its possibilities and is developing appropriate examples to demonstrate these options. The thesis is complemented by animations. These animations are demonstrating how to work in Scratch.

At first the thesis is defining what Scratch is and describes its history. The user interface was divided to several parts. These parts were analyzed and described. The possibilities of usage are defined and described. The thesis also mentions the large Scratch user community.

In the last part of the thesis several examples of usage are introduced. Each example consists of task, goals and the suggestion of solution.

SEZNAM LITERATURY

- [1] Scratch Statistics - Vymysli, programuj, poděl se. *Scratch Statistics* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: <http://scratch.mit.edu/statistics/>
- [2] Scratch Wiki. *Development of Scratch 1.0* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: http://wiki.scratch.mit.edu/wiki/Development_of_Scratch_1.0
- [3] Scratch Wiki. *Scratch Timeline* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: http://wiki.scratch.mit.edu/wiki/Scratch_Timeline
- [4] Scratch Wiki. *Scratch 1.0* [online]. 2014 [Citace 28. 6. 2015]. Dostupné z: http://wiki.scratch.mit.edu/wiki/Scratch_1.0
- [5] Scratch Wiki. *Experimental Blocks* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: http://wiki.scratch.mit.edu/wiki/Experimental_Blocks
- [6] Scratch Wiki. *Scratch 1.4* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: http://wiki.scratch.mit.edu/wiki/Scratch_1.4
- [7] Scratch Wiki. *Motor Blocks* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: http://wiki.scratch.mit.edu/wiki/Motor_Blocks
- [8] Scratch Wiki. *Scratch Modification* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: http://wiki.scratch.mit.edu/wiki/Scratch_Modification
- [9] Scratch Wiki. *List of Scratch Modifications* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: http://wiki.scratch.mit.edu/wiki/List_of_Scratch_Modifications
- [10] Scratch Wiki. *Bingo (Scratch Modification)* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: http://wiki.scratch.mit.edu/wiki/Bingo_%28Scratch_Modification%29
- [11] Scratch Wiki. *Build Your Own Blocks (Scratch Modification)* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: http://wiki.scratch.mit.edu/wiki/Build_Your_Own_Blocks_%28Scratch_Modification%29
- [12] Snap!. *Snap! (Build Your Own Blocks) 4.0* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: <http://byob.berkeley.edu/>
- [13] Scratch Wiki. *Scratch 2.0* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: http://wiki.scratch.mit.edu/wiki/Scratch_2.0
- [14] Scratch Wiki. *Scratch 2.0 Offline Editor* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: http://wiki.scratch.mit.edu/wiki/Offline_Editor
- [15] Scratch. *Scratch 2.0 Offline Editor* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: <https://scratch.mit.edu/scratch2download/>
- [16] Scratch Wiki. *Hidden Features* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: http://wiki.scratch.mit.edu/wiki/Hidden_Features#Turbo_Mode
- [17] MCMANUS, Sean. *Scratch programming in easy steps*. Leamington Spa, Warwickshire, U. K: In Easy Steps, 2013. ISBN 9781840786125.
- [18] Scratch Wiki. *Scratch Wiki* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: <http://wiki.scratch.mit.edu/wiki/Wiki>

- [19] Scratch Wiki. *Scratch Wiki:Become a contributor* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: http://wiki.scratch.mit.edu/wiki/Scratch_Wiki:Become_a_contributor
- [20] Scratch - Pro učitele. *Pro učitele* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: <https://scratch.mit.edu/educators/>
- [21] Scratch Wiki. *ScratchEd* [online]. 2014 [Citace 28. 6. 2015]. Dostupné z: <http://wiki.scratch.mit.edu/wiki/ScratchEd>
- [22] Scratch Wiki. *ScratchJr* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: <http://wiki.scratch.mit.edu/wiki/ScratchJr>
- [23] App Annie. *ScratchJr* [online]. 2015 [Citace 28. 6. 2015]. Dostupné z: <https://www.appannie.com/apps/ios/app/scratchjr/>

SEZNAM OBRÁZKŮ, TABULEK, GRAFŮ A DIAGRAMŮ

Obrázek 1: Rozložení věku nových uživatelů [1]	5
Obrázek 2: Scratch Timeline	6
Obrázek 3: <i>Scratch 2.0</i> online editor	9
Obrázek 4: Webová stránka s odkazy ke stažení [15]	11
Obrázek 5: Instalace Scratch 2.0	11
Obrázek 6: Prostředí <i>Scratch 2.0</i>	12
Obrázek 7: Paleta nástrojů	14
Obrázek 8: Souřadnice Spritu	15
Obrázek 9: Ovládací prvky	15
Obrázek 10: nabídka s možnostmi - blok	16
Obrázek 11: nabídka s možnostmi - skriptovací plocha	16
Obrázek 12: okno komentáře	16
Obrázek 13: Porovnání položek menu <i>Soubor</i>	22
Obrázek 14: Demonstrace tvaru bloků	23
Obrázek 15: Prostředí ScratchJR [23]	31
Obrázek 16: Třídění odpadu	33
Obrázek 17: Bludiště	34
Obrázek 18: Skákání na platformy	36
Obrázek 19: Střílení na netopýry	37

PŘÍLOHY

Přílohy se nacházejí na přiloženém DVD:

- Praktické příklady vytvořené v programu Scratch
- Animace
- Elektronická verze bakalářské práce