

Úvod do jazyka Scratch

V předchozích kapitolách této knihy jste získali hodně informací o tom, jak používat programy, které pro počítač Raspberry Pi napsali jiní lidé. Hlavním cílem projektu Raspberry Pi je však naučit uživatele, aby psali své vlastní programy – a nevztahuje se to jen na dospělé. Nadace Raspberry Pi Foundation usiluje o to, aby se její počítač rozšířil jako vzdělávací nástroj pro všechny věkové skupiny.

K dosažení tohoto cíle je klíčové, aby se tvorbou vlastního softwaru mohly bavit i malé děti a nemusely jen konzumovat kód, který vytvořil někdo jiný. Hlavním nástrojem je přitom jazyk *Scratch*.

Představení jazyka Scratch

Jazyk Scratch vznikl roku 2006 ve skupině Lifelong Kindergarten laboratoře Media Lab Massachusettského technického institutu (MIT). Tento jazyk, odvozený od jazyků Squeak a Smalltalk, je založen na klíčových programátorských principech, které zpřístupňuje všem zájemcům. Místo psaní dlouhých řetězců, které je pro mladší děti únavné a nudné, se využívá jednoduché prostředí, v němž lze skládat stavební kameny přetažením myši. Přesto však rozvíjí programátorské myšlení a uvádí základní koncepce, které se používají ve všech programovacích jazycích.

Programovací prostředí je oficiálně určeno uživatelům od osmi let věku, ale s trochou pomoci a vedení je přístupné dokonce i mladším programátorům. Jazyk Scratch je překvapivě silný. Za barevným a myši ovládaným uživatelským prostředím se skrývá programovací jazyk, který obsahuje působivé multimediální funkce. Nemělo by vás překvapit, že většinu z více než 2,5 milionu projektů, které uživatelé jazyka Scratch sdílejí na jeho oficiálním webu, představují hry.

Když se děti motivují k tomu, aby se učily tvořit své vlastní hry, lze do jejich zábavy vpašovat i trochu učení. Díky přátelskému uživatelskému rozhraní a vynikajícímu zpracování klíčových programovacích principů by se nemělo stát, že děti odradí náhlé nepřekonatelné problémy. Navíc koncepce osvojené v jazyce Scratch poskytují vynikající základ pro přechod na pružnější jazyky, jako je Python (viz kapitola 11, „Úvod do jazyka Python“).

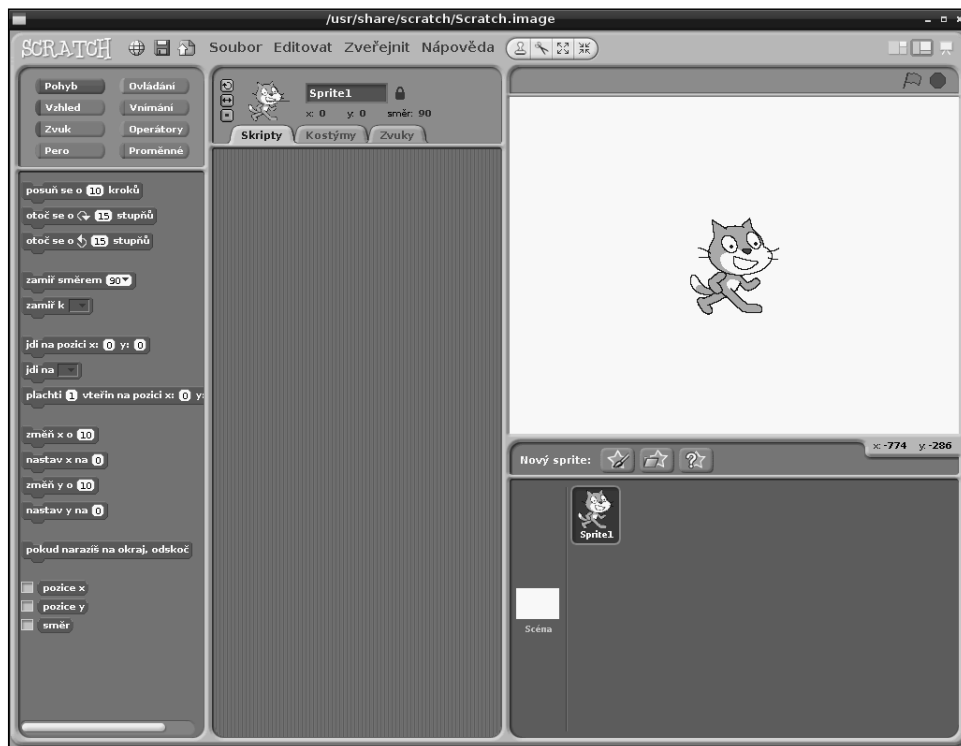
Dokonce i před přechodem na jiný jazyk nabízí jazyk Scratch více než pouhou architekturu na vývoj her. Umožňuje vytvářet interaktivní prezentace a komiksy a díky přídavnému hardwaru, jako jsou robotické sady PicoBoard a LEGO WeDo, také rozhraní s externími senzory a motory.

Doporučená distribuce Debian pro počítač Raspberry Pi standardně obsahuje nejnovější verzi vývojového prostředí Scratch, takže pokud jste postupovali podle pokynů v předchozích částech knihy, můžete hned začít. Jestliže používáte jiný operační systém, navštivte oficiální web jazyka Scratch na adrese <http://scratch.mit.edu>, kde je dostupný software ke stažení. Na stejném místě můžete najít i verze pro systémy Windows a OS X společnosti Apple.

Příklad 1: Ahoj Raspberry

Při seznamování s novým programovacím jazykem se obvykle začíná velmi jednoduchým programem, který vypisuje jeden řádek textu. Obvykle se označuje jako program „Hello, World“ (u nás často „Ahoj, světe“) a představuje první krok na cestě k tvorbě vlastních programů. Vy vytvoříte alternativní program Ahoj, Raspberry.

Oproti tradičním programovacím jazykům neočekává jazyk Scratch, že si bude uživatel pamatovat názvy instrukcí jako `print` či `inkey$`. Místo toho lze téměř vše provést přetažením bloků kódu a jejich uspořádáním do logického schématu.

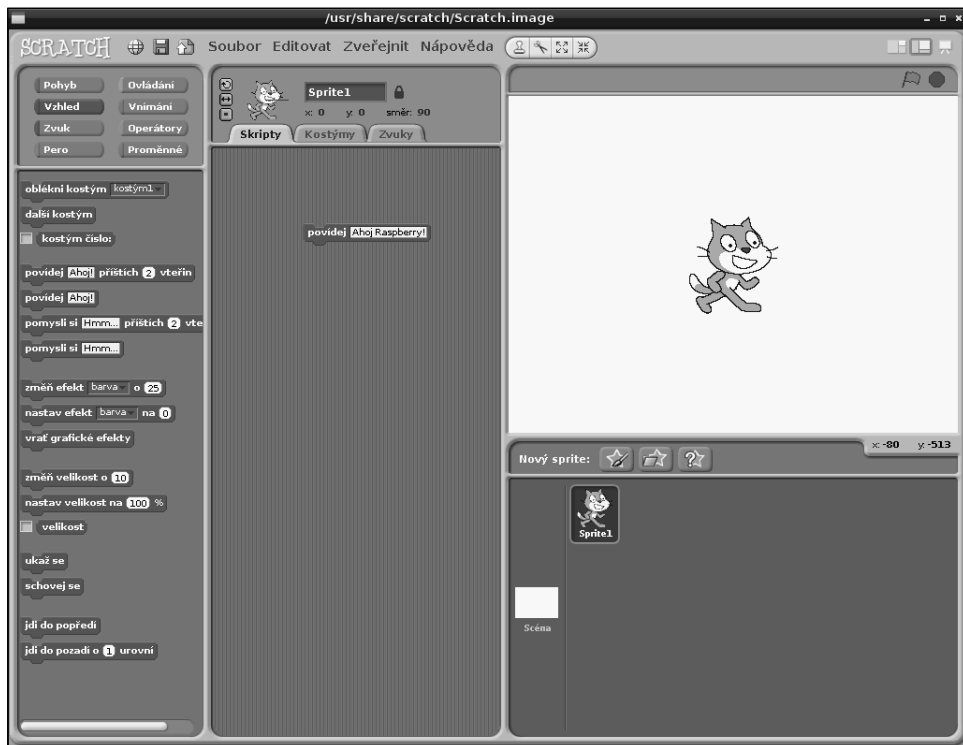


Obrázek 10.1: Hlavní rozhraní Scratch, které je spuštěno v počítači Raspberry Pi

Chcete-li začít, spusíte prostředí Scratch poklepnáním na jeho ikonu na ploše nebo klepnutím na příslušnou položku v nabídce Programming (Programování). Po několika sekundách se zobrazí hlavní rozhraní Scratch (viz obrázek 10.1). Pokud okno neleží uprostřed obrazovky nebo je příliš malé, klepněte na tlačítko Maximise (prostřední ze tří tlačítek ovládání okna na pravé straně horního okna), aby okno vyplnilo celou obrazovku.

Rozhraní Scratch se skládá z více podoken. Na levé straně je *paleta dílků* (block palette), kde jsou umístěny všechny dílky kódu, z nichž lze vytvářet programy. Vpravo dole se zobrazuje seznam objektů v programu, kterým se říká *postavy* (sprite), spolu s ovládáním *scény* (stage), na které se postavy objevují. Samotná scéna leží v pravé horní části okna a právě tam se vykresluje spuštěný program. Nakonec prostřední část okna slouží ke stavbě vlastního programu.

Aby mohl uživatel rychle začít, nový projekt Scratch již obsahuje prázdnou scénu a jednu postavu. Zatím však chybí program, takže se po klepnutí na zelenou ikonu praporeku v pravé horní části okna nic nestane, protože prostředí Scratch dosud neví, co chcete provést.



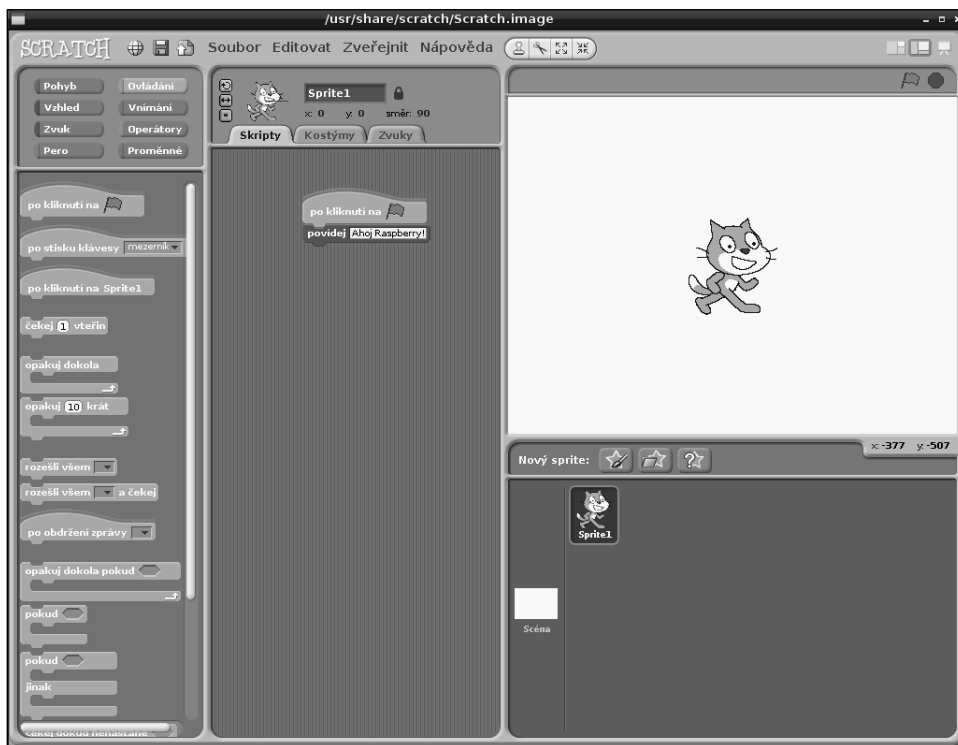
Obrázek 10.2: První dílek umístěný do programu Scratch

U programu Ahoj, Raspberry je nutné přepnout paletu dílků na levé straně obrazovky do režimu Looks (Vzhled) klepnutím na jeho tlačítko. Níže v seznamu dílků Looks (Vzhled) se nachází dílek s popisem say Hello! (povídej Ahoj!). Klepněte na tento dílek a přetáhněte jej na prázdné místo uprostřed podokna označeného Scripts (Skripty). Chcete-li se držet tradičního

názvu programu, který vznikl již před desítkami let, můžete také případně klepnout na popis Hello! (Ahoj!) dílku a upravit jej tak, aby obsahoval popis Ahoj , Raspberry! (viz obrázek 10.2).

Ani po dalším klepnutí na zelený praporek program nic neudělá. Prostředí Scratch sice ví, že má postava kocoura něco říci, ale zatím neví kdy. Událost vyžaduje *aktivační dílek* (trigger block), který najdete v části Control (Ovládání) palety dílků.

Přejděte do této části klepnutím na tlačítko Control (Ovládání) a poté přetáhněte myší první položku označenou when [flag icon] clicked (po kliknutí na [ikonu praporeku]) a umístěte ji těsně nad fialový dílek say (povídej – viz obrázek 10.3). Pokud dílek přetáhnete dostatečně blízko, měl by se automaticky připojit ke stávajícímu dílku jako kousek skládačky.

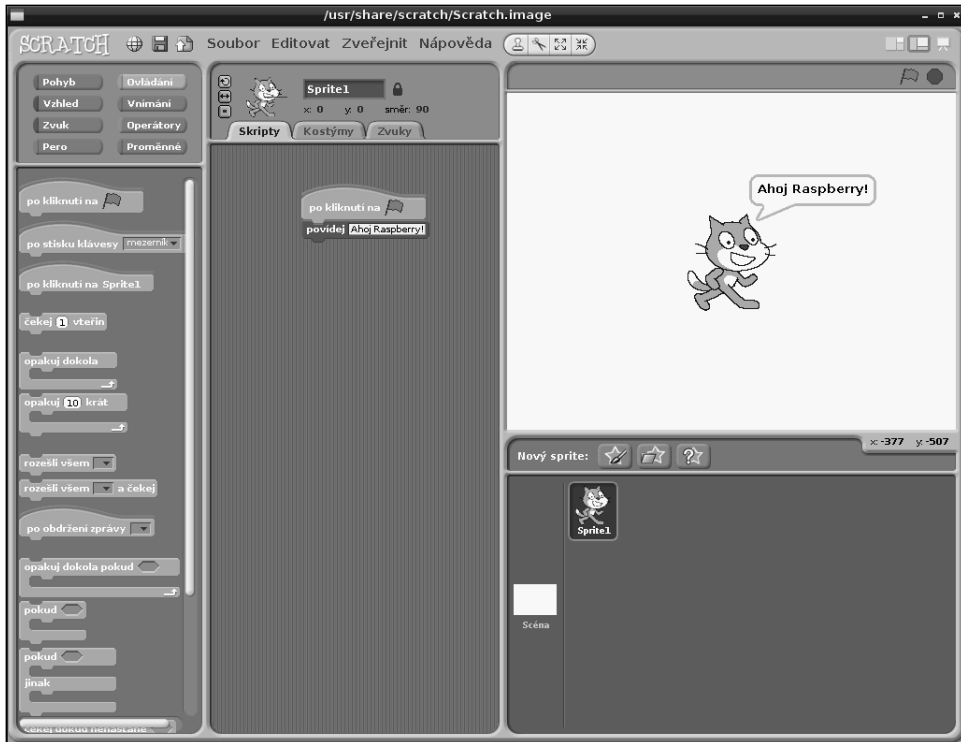


Obrázek 10.3: Dílek Control (Ovládání) připojený k dílku Looks (Vzhled) v prostředí Scratch

Toto propojování více dílků dohromady představuje základní princip jazyka Scratch. Podíváte-li se na dílek Control (Ovládání), který jste právě umístili, můžete si všimnout, že ve své horní části nemá žádnou spojovací díru. To znamená, že nad něj nemůžete uložit další dílek, protože tento dílek Control (Ovládání) má přímo aktivovat řadu jiných dílků a musí se nacházet na začátku řetězce. Ve své dolní části je však dílek say (povídej) vybaven konektorem, který zapadá do horní části jiných dílků. Z toho lze poznat, že lze pod něj umístit více dílků.

Když jsou dva dílky na svém místě, znovu klepněte na ikonu zeleného praporkku v pravém horním rohu obrazovky. Tentokrát se u kocourových úst objeví textová bublina (viz obrázek 10.4) a program Ahoj, Raspberry je hotový.

Než se pustíte do dalšího příkladu, nezapomeňte svou práci uložit pomocí nabídky File (Soubor). Prostředí Scratch je navrženo tak, že může být v jednu chvíli otevřen pouze jediný projekt. Jestliže tedy vytvoříte nový prázdný soubor, bude aktuální soubor uzavřen, aby udělal místo pro nový. Nemusíte se však obávat, že na to při přechodu k novému souboru zapomenete. Pokud jsou v existujícím souboru neuložené změny, prostředí Scratch vás před uzavřením souboru vyzve k tomu, abyste jej uložili.



Obrázek 10.4: Program Ahoj Raspberry spuštěný v prostředí Scratch

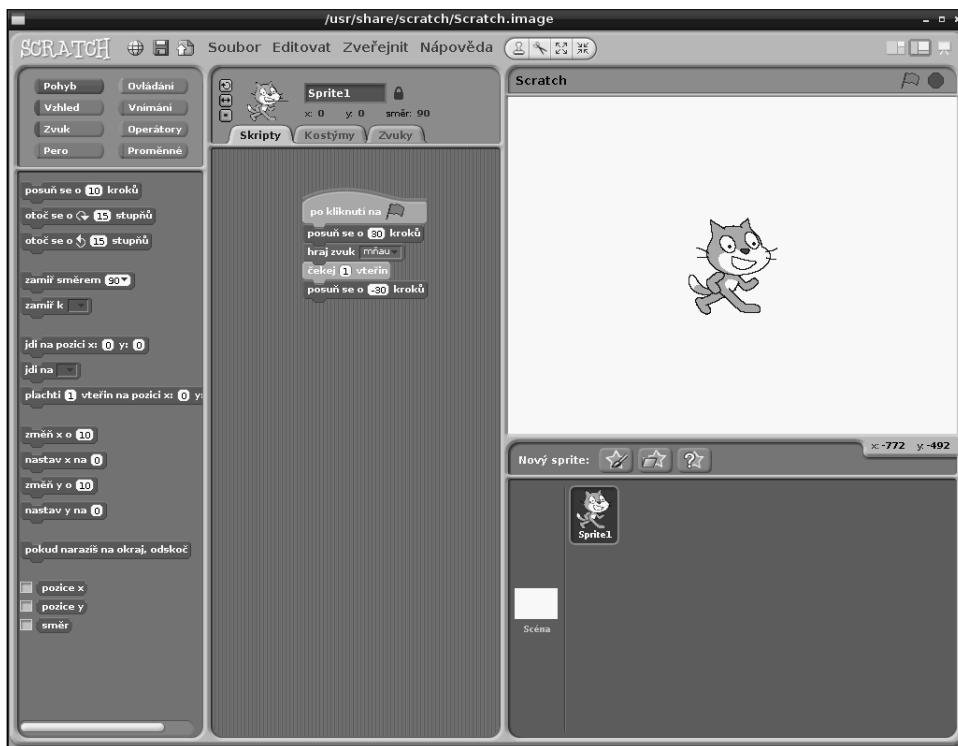
Příklad 2: Animace a zvuk

Program Ahoj Raspberry sice představuje klasický příklad, ale není příliš zajímavý. Neukazuje také skutečné možnosti jazyka Scratch, které leží v jeho působivých multimediálních možnostech a systému ovládání postav. Tento systém se hodí zejména pro jednoduché animace, které mohou tvořit základ interaktivní hry.

Nejdříve začněte nový projekt jazyka Scratch. Přitom můžete buď znovu spustit celé prostředí, nebo zvolit příkaz New (Nový) z nabídky File (Soubor). Jako u každého nového projektu poskytnete prostředí Scratch výchozí postavu. Jedná se o postavu, kterou budete řídit.

Jednoduchou animaci v prostředí Scratch lze ovládat pomocí části Motion (Pohyb) palety dílků. Při spuštění nového projektu je tato paleta nastavena jako výchozí. Přetáhněte dílek označený `move 10 steps` (posuň se o 10 kroků) do oblasti Scripts (Skripty). Jak je zřejmé z jeho názvu, tento dílek požádá vybranou postavu, aby se posunula o 10 kroků ve směru, kam je aktuálně otočena. Výchozí postava jazyka Scratch je vždy orientována přímo doprava. Dílek `move 10 steps` (posuň se o 10 kroků) tedy zajistí posun postavy o 10 kroků doprava.

Deset kroků je poměrně málo. Klepněte tedy na hodnotu 10 a změňte ji na 30. Popis dílku by se měl změnit na `move 30 steps` (posuň se o 30 kroků). Animace kocoura, který se pouze pohybuje po scéně zleva doprava, však není příliš zajímavá. Přepněte tedy na paletu dílků Sound (Zvuk) a přetáhněte dílek `play sound meow` (hraj zvuk mňau) do oblasti Scripts (Skripty) a připojte jej pod existující dílek `move` (posuň se). Chcete-li kocoura chvíli ponechat v této pozici, přidejte z palety dílků Control (Ovládání) dílek `wait 1 secs` (čekej 1 vteřin). Bez tohoto dílku by postava kocoura rychle přeskočila ze své počáteční pozice do cílové.

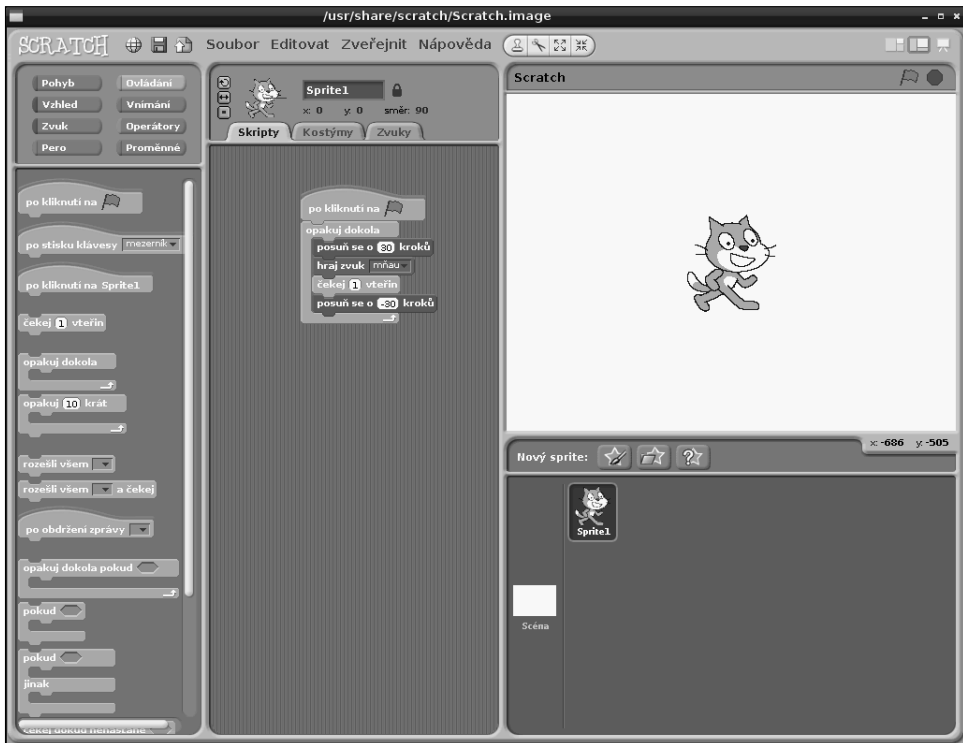


Obrázek 10.5: Dokončený program animace v jazyce Scratch

Pokud chcete nastavit, aby bylo možné program spustit vícekrát a postava kocoura přitom nezmizela za okraj scény, doplňte další dílek `move 10 steps` (posuň se o 10 kroků) pod dílek `play sound` (hraj zvuk) a upravte jej tak, aby měl popis `move -30 steps` (posuň se o -30 kroků). Prostředí Scratch bez problémů umožňuje zadávat taková záporná čísla: jestliže hodnota 30 způsobí, že se postava posune o určitou vzdálenost doprava, pak hodnota -30 zajistí posun postavy o přesně stejnou vzdálenost, ale v opačném směru.

Nakonec z palety dílků Control (Ovládání) přidejte do horní části oblasti Scripts (Skripty) dílek `when [flag icon] clicked` (po kliknutí na [ikonu praporku]). Dokončený program je znázorněn na obrázku 10.5. Program lze spustit klepnutím na ikonu zeleného praporku v pravém horním rohu okna. Kvůli silnějšímu efektu nezapomeňte k počítači Pi připojit sluchátka nebo reproduktory!

Tento jednoduchý animační program je možné různými způsoby rozšiřovat. Pomocí možnosti `New Sprite` (Nový sprite) přímo pod scénou na pravé straně okna prostředí Scratch může programátor přidávat další postavy, které se dokáží pohybovat a přehrávat zvuky nezávisle na sobě. Přidáte-li dílek `say` (povídej) z prvního příkladu (nebo podobný dílek `think` (pomysli si), který místo mluvené bubliny vytvoří myšlenou bublinu), můžete vytvořit animovaný komiks.



Obrázek 10.6: Přidání nekonečného cyklu do jednoduché animace jazyka Scratch

Důležitější je, že dokonce i tento jednoduchý příklad učí důležité programátorské principy: navzdory své délce pouhých pěti dílků zahrnuje pohyb postavy v kladných i záporných souřadnicích, přehrávání zvuku a koncepci čekání (delay) v programu. Pokud se nebojíte, že se zblázníte z opakovaně přehrávaného zvuku, můžete vyzkoušet další koncepci a přidat z palety Control (Ovládání) dílek forever (opakuj dokoła – viz obrázek 10.6). Tento dílek do programu dodá cyklus (loop), takže program bude seznamem dílků procházet stále znovu a znovu – alespoň do té doby, než zvuk začne být nesnesitelný a klepnete na červené tlačítko stop v horní části scény!

Příklad 3: Jednoduchá hra

V jazyce Scratch lze sice vytvářet jednoduché animace, ale software umožňuje i tvorbu interaktivních programů, které reagují na vstup z klávesnice. Když výše popsany program doplníte o několik jednoduchých animačních ovladačů, můžete získat jednoduchou hru – a zároveň si osvojit princip *kolize postav* (sprite collision), *příkazů if a vstupu*.

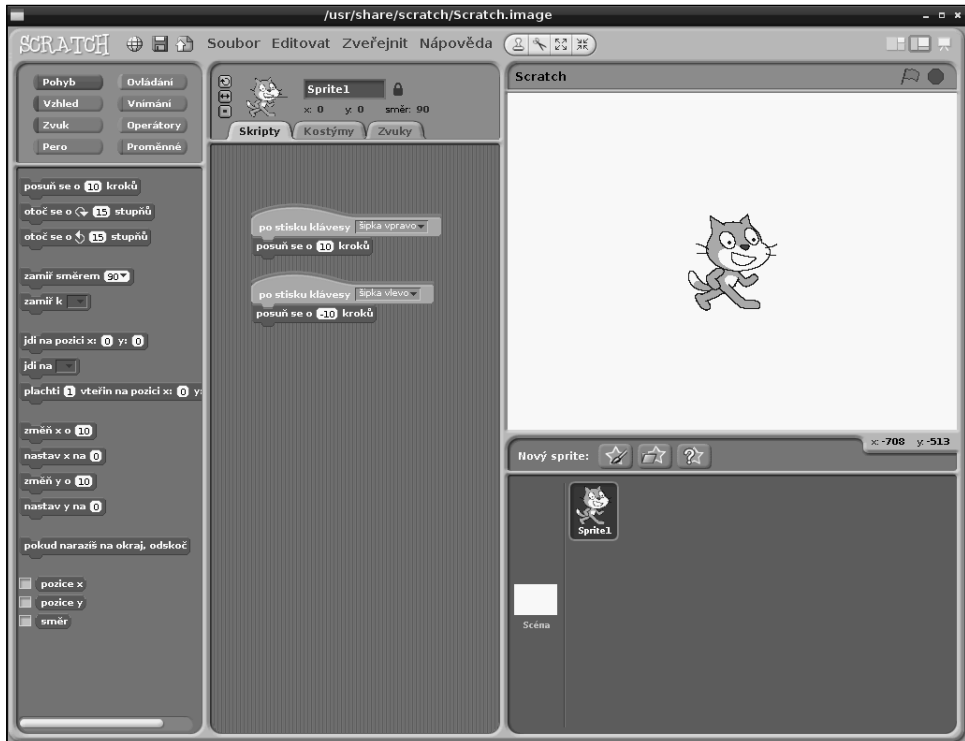
V tomto příkladu začnete nový projekt jazyka Scratch (pokud jste to zatím neudělali, nezapomeňte uložit předchozí příklad). Nejdříve do oblasti Scripts (Skripty) přetáhněte dílek move 10 steps (posuň se o 10 kroků). Místo toho, abyste nastavili spuštění dílků kódu při klepnutí na ikonu praporku, přejděte tentokrát do palety dílků Control (Ovládání) a přetáhněte dílek when space key pressed (po stisku klávesy mezerník) nad dílek move (posuň se).

Jak je zřejmé z jeho názvu, dílek when space key pressed (po stisku klávesy mezerník) sleduje vstup od uživatele – v tomto případě stisknutý mezerník – a používá jej jako aktivátor spuštění seznamu dílků. Dílek funguje kdykoli – jestliže stisknete mezerník ihned, postava splní pokyn a posune se o 10 kroků doprava.

Dílek when space key pressed (po stisku klávesy mezerník) se také od dílku when [flag icon] clicked (po kliknutí na [ikonu praporku]) liší dalším důležitým způsobem: lze jej upravit. Po klepnutí na šipku dolů vedle slova space (mezerník) se zobrazí seznam všech klíčů, které může dílek monitorovat. Ze seznamu poté vyberte položku right arrow (šipka vpravo). Dílek se tím změní na when right arrow key pressed (po stisku klávesy šipka vpravo).

Hra, ve které se hráč může pohybovat pouze v jednom směru, není příliš zábavná, takže do oblasti Scripts (Skripty) přetáhněte nový dílek when space key pressed (po stisku klávesy mezerník). Tento dílek nelze propojit s existujícím seznamem dílků, protože aktivační dílek může být pouze jeden. Začněte tedy nový seznam o něco níže. Stejně jako předtím použijte tlačítko se šipkou dolů vedle slova space (mezerník), abyste dílek přizpůsobili. Změňte jej na dílek when left arrow key pressed (po stisknutí klávesy šipka vlevo). Nakonec přepněte paletu dílků zpět do režimu Motion (Pohyb) a připojte dílek move 10 steps (posuň se o 10 kroků) pod nový dílek when left arrow key pressed (po stisknutí klávesy mezerník). Potom změňte popis dílku na move -10 steps (posuň se o -10 kroků).

Stisknete-li nyní levou a pravou šipku, zjistíte, že se kocour pohybuje podle vstupu z klávesnice (viz obrázek 10.7): při stisknutí šipky doleva se kocour posune o 10 kroků vlevo (ačkoli z hlediska jazyka Scratch se jedná o posun o minus 10 kroků vpravo) a při stisknutí šipky vpravo se kočka přemístí o 10 kroků vpravo.



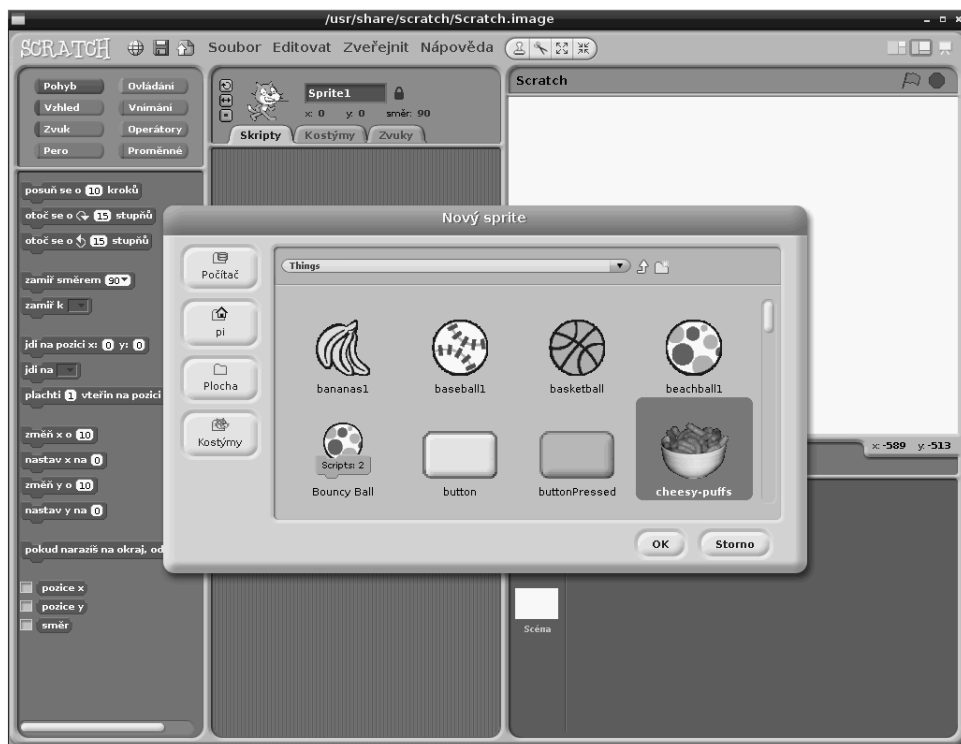
Obrázek 10.7: Ovládání pohybu pohyblivého prvku jazyka Scratch pomocí vstupních dílků

Když nyní víte, jak může hráč ovládat postavu, je potřeba přiřadit jí nějakou funkci. Protože se jedná o velmi jednoduchou hru, bude cíl také prostý: najít nějaké jídlo. Klepněte na tlačítko Choose New Sprite From File (Výběr nového sprite ze souboru), které je prostřední ze tří tlačítek nad seznamem postav v pravé dolní části okna Scratch.

Zobrazí se dialogové okno s pokynem, abyste vybrali postavu: poklepejte na složku Things (Věci) a poté poklepejte na pohyblivý postavu Cheesy-Puffs (kukuřičné křupky se sýrem – viz obrázek 10.8). Postava se objeví jako nová položka seznamu postav a získáte tak nový objekt, který můžete ve hře kontrolovat.



Tip: Jazyk Scratch přirozeně podporuje *multithreading* a částečně je *objektově orientován*. To znamená, že ke každému objektu v programu (včetně postav) lze připojit vlastní kód a všechny dílky kódu mohou fungovat souběžně a nezávisle na jiných dílcích. Když tyto možnosti vhodně uplatníte, můžete díky nim vytvářet poměrně složité programy.



Obrázek 10.8: Přidání postavy Cheesy-Puffs do projektu hry jazyka Scratch

Každá postava přidaná do projektu Scratch se standardně zobrazí přesně uprostřed scény. Z tohoto důvodu zakryje stávající postavu kocoura. Chcete-li tento problém vyřešit, klepněte na novou postavu a přetáhněte ji doprava od kocoura.

Přesunutá mísa kukuřičných křupek je v poměru k antropomorfnímu dvounohému kocourovi stále příliš velká. Klepněte tedy na tlačítko Shrink Sprite (Zmenšit sprite), které je umístěno v pravé horní části oblasti scény a zobrazuje čtyři šipky, které míří dovnitř. Jestliže si nejste jisti, o které tlačítko se jedná, můžete umístit ukazatel myši nad jednotlivé ikony a zobrazí se stručný popis jejich účelu.

Klepnete-li na tlačítko Shrink Sprite (Zmenšit sprite) nebo na tlačítko Grow Sprite (Zvětšit sprite), které má přesně opačnou funkci, ukazatel myši se změní na kopii ikony tlačítka. Pomocí tohoto nového ukazatele klepněte na postavu Cheesy-Puffs, abyste ji zmenšili. Při každém následujícím klepnutí se bude postava dále zmenšovat. Jakmile její velikost bude vypadat přiměřeně, klepněte na kterékoli místo mimo oblast scény, abyste obnovili normální podobu ukazatele myši. Chcete-li, můžete pak mísu přesunout blíže k pravému okraji scény.

Nyní se pokuste přesunout postavu kocoura směrem k postavě Cheesy-Puffs. Jak se sami přesvědčíte, při setkání obou postav se nic nestane. To proto, že program neobsahuje žádné poky-

ny, co má dělat při překrytí dvou postav, které se označuje jako *kolize postav*. Neprovede tedy žádnou akci. To lze napravit pomocí nového typu dílku: dílku *Sensing* (Vnímání).

Pokud je postava Cheesy-Puffs aktivní, měl by se její obrázek objevit v horní části podokna Scripts (Skripty). V opačném případě stačí poklepat na prvek na scéně. Při aktivní postavě Cheesy-Puffs přepněte paletu dílků do režimu Sensing (Vnímání) klepnutím na stejnojmenné tlačítko. Z palety Sensing (Vnímání) přetáhněte do podokna Scripts (Skripty) dílek *touching ?* (dotýká se ?).

Stejně jako dílek *when space key pressed* (po stisku klávesy mezerník), kterým jste kontrolovali pohyb postavy kocoura, lze i dílek *touching ?* (dotýká se ?) upravit. Klepněte na tlačítko se šipkou dolů vedle otazníku a ze seznamu zvolte položku *Sprite1* (postava kocoura). Tento dílek se nyní bude aktivovat při kolizi dvou postav.



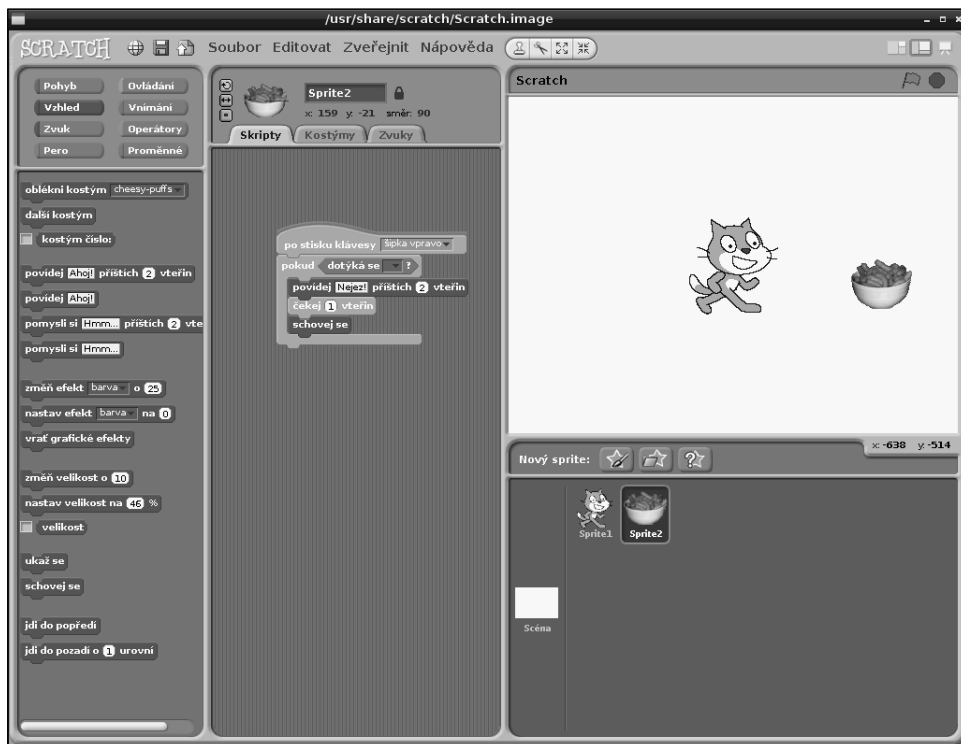
Tip: Chcete-li postavu pojmenovat, klepněte v podokně Scripts (Skripty) na rámeček vedle obrázku postavy a zadejte název. Když postavám dáte srozumitelná jména (například *Kocour*, *Kukuři čně-Křupky* atd.) bude později mnohem snazší sledovat, co se v programu děje.

Podívejte se na tvar dílku *touching Sprite1?* (dotýká se Sprite1?). Je zřejmé, že nahoře ani dole nemá žádné propojovací výstupky a svým tvarem připomíná diamant. Stejný tvar slouží ve vývojových diagramech jako rozhodovací bod. To není náhoda: většina dílků Sensing (Vnímání) funguje teprve po svém vložení do dílku Control (Ovládání).

Přepněte paletu dílků do režimu Control (Ovládání) a vyhledejte dílek *if* (pokud) – vypadá jako rozmáčkklé a hrbolaté písmeno C. Všimněte si, že dílek *if* (pokud) má výřez ve tvaru diamantu, což je tvar dílku *touching Sprite1?* (dotýká se Sprite1?). Přetáhněte dílek *if* (pokud) do podokna Scripts (Skripty) a poté přetáhněte dílek *touching Sprite1?* (dotýká se Sprite1?) na jeho výřez ve tvaru diamantu. Získáte dvoubarevný dílek, který má aktuálně popis *if touching Sprite1?* (dotýká se Sprite1?).

Tento kombinovaný dílek zastupuje v programu *podmínku*: po dosažení tohoto bodu programu je veškerý kód umístěný v rámci podmínky proveden tehdy a pouze tehdy, je-li *podmínka* splněna. V tomto případě se jedná o podmínku, že se postava *Sprite2* dotkla postava *Sprite1*. Pomocí logických dílků *and* (a), *or* (nebo) a *not* (není) z palety dílků Operators (Operátory) lze realizovat poměrně složité scénáře.

Z palety dílků Looks (Vzhled) přetáhněte dílek *say Hello! For 2 secs* (povídej Ahoj! přístých 2 vteřin) doprostřed podmínky *if touching Sprite1?* (pokud se dotýká Sprite1?). Změňte text na *Nejez!* a poté přidejte dílek Control (Ovládání) s názvem *wait 1 secs* (čkej 1 vteřin) a změňte hodnotu na 2. Nahoru doplňte dílek *when space key pressed* (po stisku klávesy mezerník) a upravte jeho popis na *when right arrow key pressed* (po stisku klávesy šipka vpravo). Nakonec na konec cyklu přetáhněte dílek *hide* (schovej se) z palety Looks (Vzhled). Dostanete seznam dílků, který je znázorněn na obrázku 10.9.



Obrázek 10.9: Řízení postavy Cheesy-Puffs pomocí dílku if (pokud) jazyka Scratch

Booleovská logika

Booleovská logika či booleovská algebra, která svůj název získala podle George Booleho, představuje klíčový princip fungování počítačů. V jazyku Scratch je booleovská logika implementována ve třech dílcích Operators (Operátory): **and (a)**, **or (nebo)** a **not (není)**.

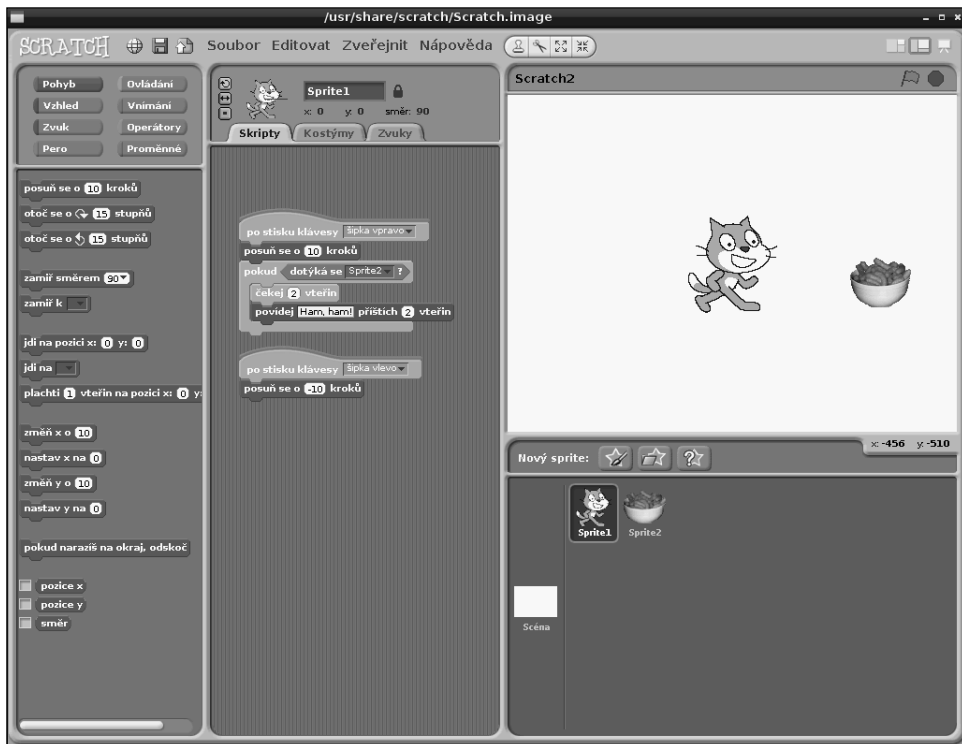
Pokud má být pravdivý výstup operátoru **and (a)**, je nutné, aby byly pravdivé oba jeho vstupy – v případě jazyka Scratch se jedné o dílky Sensing (Vnímání). Jestliže není pravdivý některý z jeho vstupů nebo oba vstupy, bude nepravdivý i jeho výstup. Pomocí tohoto operátoru můžete například zkontrolovat, zda se postava dotýká dvou jiných postav současně.

Operátor **or (nebo)** vyžaduje, aby byl pravdivý některý z jeho dvou vstupů. Je-li pravdivý alespoň jeden vstup, bude pravdivý i výstup operátoru. Nabízí se tím praktický způsob opakovaného použití kódu: máte-li více postav, kterých se postava hráče nesmí dotknout, stačí použít jediný dílek kódu s operátorem **or (nebo)**, který se aktivuje při kontaktu s libovolnou nepřátelskou postavou.

Poslední operátor **not (není)** se označuje jako *invertor*: ať už má jakýkoli vstup, jeho výstup je vždy opačný. Jestliže je jeho vstup nepravdivý, výstup je pravdivý. Naopak platí, že pokud je vstup pravdivý, výstup je nepravdivý.

Poklepáním na postavu kocoura na scéně přejdete zpět k úpravám jeho skriptů. Skript vytvořený pro postavu Cheesy-Puffs zmizí, ale nelekejte se. Zůstává zachován, ale znovu se objeví teprve tehdy, začnete-li upravovat příslušnou postavu.

Přetáhněte další dílek `if` (pokud) z palety Control (Ovládání) spolu s dalším detekčním dílkem `touching ?` (dotýká se ?). Tentokrát změňte dílek Sensing (Vnímání) tak, aby oba dílky společně měly popis `if touching Sprite2?` (pokud dotýká se Sprite2?). Do tohoto dílku vložte dílek Control (Ovládání) `wait 1 secs` (čkej 1 vteřin), jehož hodnotu změňte na 2, a dílek Looks (Vzhled) `say Hello! for 2 secs` (povídej Ahoj! příštích 2 vteřin), kde původní text nahraďte textem `Ham, ham!`. Nakonec přetáhněte celou sadu dílků nahoru, abyste ji propojili s dolní částí existujícího dílku `when right arrow key pressed` (po stisku klávesy šipka vpravo) pod dílkem `move 10 steps` (posuň se o 10 kroků). Hotový kód pro postavu kocoura by měl odpovídat dílkům na obrázku 10.10.



Obrázek 10.10: Dokončený kód jednoduché hry jazyka Scratch s pohyblivým prvkem kočky

Když pomocí klávesy se šipkou doprava posunete kocoura směrem k postavě Cheesy-Puffs, hra začíná. Když se kocour dostane k postavě Cheesy-Puffs, proběhne rozhovor a mísa křupek by měla zmizet.

Tento příklad sice efektivně představuje několik důležitých programátorských přístupů, ale rozhodně nejde o optimální způsob, jak by takovou hru bylo možné naprogramovat. Jazyk

Scratch obsahuje systém *vysílání zpráv* (message broadcast), který kódu připojenému k jednomu objektu umožňuje komunikovat s kódem připojeným k jinému objektu. Díky tomu lze dosahovat mnohem elegantnějších výsledků kolizí, které nemusí spoléhat na pečlivě načasované pauzy, aby dávaly smysl.

Chcete-li experimentovat s vysíláním zpráv, vyzkoušejte dílky broadcast (rozešli všem) a when I receive (po obdržení zprávy) z palety Control (Ovládání). Zpráva vytvořená pro dílek vysílání v libovolném objektu může aktivovat kód v kterémkoli jiném objektu pomocí příznaku when I receive (po obdržení zprávy). To znamená, že tímto způsobem lze vzájemně propojit více objektů a jejich kód.

Robotika a senzory

Kromě čistě softwarových vstupů a výstupů je také možné navázat programy jazyka Scratch na vnější hardware pomocí senzoricích desek a robotických systémů. Díky své multithreadingové povaze a výkonnému systému zaslání zpráv mezi procesy lze pomocí jazyka Scratch vytvořit překvapivě pokročilý robotický modul. Jedná se o zdaleka nejjednodušší způsob, jak z počítače Raspberry Pi udělat centrum jednoduchého robota.

Na rozdíl od programovacího jazyka Python, který je sice výkonnější, ale značně složitější, nedokáže jazyk Scratch komunikovat s externím hardwarem pomocí portu GPIO počítače Raspberry Pi. Aby mohl interagovat s fyzickým světem, je tedy nejdříve nutné jej doplnit o dodatečné zařízení.

Detekce pomocí sady PicoBoard

Sada PicoBoard, kterou navrhl výrobce elektroniky SparkFun, je rozšiřující modul, který se připojuje k počítači s prostředím Scratch. Poskytuje proměnlivý vstup ovládaný jezdcem, světelný senzor, tlačítko, zvukový senzor a čtyři páry krokosvorek, které umožňují monitorovat externí elektrická zařízení.

V současnosti jsou k dispozici dva typy sad PicoBoard: původní model, který používá sériové připojení, a jeho nástupce, který pracuje s připojením USB. Druhý model je kompatibilní s počítačem Raspberry Pi. Stačí jej připojit do volného portu USB nebo do portu rozbočovače USB připojeného k počítači Pi a restartovat prostředí Scratch. Poté lze využívat různé jeho funkce přímo z rozhraní Scratch.

Podrobnosti o sadě PicoBoard naleznete na adrese <http://www.sparkfun.com/products/10311>.

Robotika se sadou LEGO

Kromě doplňkových modulů se senzory je možné pomocí jazyka Scratch řídit i externí robotický systém. Robotická sada LEGO Education WeDo umožňuje připojit k počítači motory,

senzory vzdálenosti a senzory rychlosti. Všechny tyto prvky lze přitom řídit pomocí specializovaných dílků, které jsou dostupné v prostředí Scratch.

Dětem sada WeDo nabízí vynikající úvod do robotiky: součásti se kombinují stejně snadno jako běžné dílky LEGO a ani při nesprávném zapojení nehrozí riziko elektrického šoku. Sada WeDo je kompatibilní s běžnými sadami LEGO a LEGO Technic, takže je možné rychle a snadno budovat velké projekty.

Další informace o použití sady LEGO Education WeDo s jazykem Scratch naleznete na stránce <http://info.scratch.mit.edu/WeDo>.

Další zdroje informací

Tato kapitola sice slouží jako stručný úvod do jazyka Scratch, ale ani zdaleka toto téma nevyčerpává. Pro mladší čtenáře, kteří se obvykle učí snáze, když jejich lekce doprovází hodně barevných obrázků, je také poněkud popisná.

Sekce Support (Podpora) oficiálního webu jazyka Scratch, které provozuje univerzita MIT na adrese <http://info.scratch.mit.edu/support>, obsahuje odkaz na příručku pro začátečníky (Getting Started Guide) ve formátu PDF. Pestrá příručka vysvětluje práci s jazykem Scratch způsobem, který odpovídá dětskému čtenáři. Jedná se o skvělý výukový nástroj. Učení lze dále oživit díky kombinaci s kartami Scratch Cards, což jsou kartičky ke stažení, které obsahují definice jednotlivých typů dílků jazyka Scratch.

Univerzita MIT také provozuje fórum uživatelů jazyka Scratch, kde se mohou nadšenci učit od zkušenějších kolegů a sdílet řešení běžných problémů. Registrace je bezplatná a web dobře funguje i v prohlížeči Midori, který je dostupný v distribuci Debian pro počítač Raspberry Pi. Fóra jsou k dispozici na adrese <http://scratch.mit.edu/forums/>.

Chcete-li se však zdokonalit v jazyku Scratch, je nejjednodušší prostě si začít hrát. Název Scratch odkazuje na používání gramofonu: když diskžokej otáčí desku, jehla přejede přes drážky a ozve se zvuk škrábnutí. Tak jako diskžokejové mixují z hotových písniček nové skladby, nadšení uživatelé jazyka Scratch mohou posílat své výtvořky na oficiální web, kde si je jiní uživatelé mohou stáhnout, prozkoumat, upravit a kombinovat. Na oficiálním webu jazyka Scratch je v současnosti uloženo více než 2,5 milionu programů. Jedná se proto o dokonalý výukový zdroj, který umožňuje poznat, jak lze v jazyce Scratch vytvářet projekty, a sdílet vlastní nápady s ostatními. Seznam nejnovějších sdílených projektů je k dispozici na adrese <http://scratch.mit.edu/latest/shared>.