

ÚVOD

V posledních letech se na celém světě věnuje velká pozornost otázkám smysluplného zavádění digitálních technologií do škol. Diskutovanými tématy věnovanými obsahu školního vzdělávání se v současné době stávají *coding, computing, algoritmizace, programování, robotika*, a s nimi problematika rozvoje **informatického myšlení** (angl. *computational thinking*). Tato témata neodrážejí jen rostoucí potřebu IT specialistů na trhu práce, ale především reagují na přehodnocení dosavadního přístupu k počítačovým technologiím ve školním vzdělávání, které se dosud zaměřovalo hlavně na rozvoj základních uživatelských dovedností žáků používat počítače a jejich základní softwarové aplikace a pracovat v počítačových sítích včetně internetu. Nové přístupy k práci s digitálními zařízeními ve školách by se měly zaměřit na to, aby žáci poznávali, k čemu technologie mohou sloužit, jaké problémy pomáhají lidem řešit, aby rozuměli tomu, jak technologie fungují, aby se učili digitální technologie používat nejen jako nástroje pro rozvíjení kreativity, ale především pro poznávání, myšlení a učení.

Jedná se tak o návrat k myšlenkám Seymoura Paperta (1928-2016), jeho spolupracovníků (Alan Kay, Wally Feurzeig, Edith Ackermann, Gary Stager, Brian Harvey, aj.), jeho žáků (Uri Wilensky, Mitchel Resnick, aj.) a jeho pokračovatelů z celého světa (Andrej Blaho, Valentina Dagienė, Gerald Futschek, Celia Hoyles, Ivan Kalaš, Richard Noss, Jenny Sendova, Maciej Marek Syslo, Monika Tomcsányiová, Marta Turcsanyi-Szabó, aj.). Seymour Papert je zakladatelem konstrukcionismu jako teorie učení, „podle níž nejlepší způsob, jak zajistit tvar intelektuálních struktur, se děje prostřednictvím aktivní konstrukce něčeho mimo něčí hlavu, je to tedy něco hmatatelné, něco, co je možné sdílet, na rozdíl od konstruktionismu, jímž se rozumí učení jako budování znalostních struktur uvnitř hlavy jedince.“ (Stager, 2002).

V roce 1999 založili D. Cavallo, S. Papert a G. Stager na MIT „Constructionist Learning Laboratory“ (CLL), která nabízela dospívající mládeži model konstrukcionistického učení jako intervenční strategii pro vzdělávání opírající se o osm Papertových principů (Stager, 2002): (1) učíme se děláním (činnostmi), (2) technologie je pro učení vlastně takový „stavební materiál“, (3) myšlenka, nápad by měly fungovat v našem učení jako velká zábava, (4) důležité je učit se učit, (5) je zapotřebí vybrat si pro učení ten správný čas, (6) pro učení má velký význam chyba, protože jediným způsobem, jak chybu napravit, je pečlivě se podívat na to, co se stalo, když se něco pokazilo, (7) nejlepší lekcí, kterou můžeme my, učitelé a dospělí, dát našim žákům, je, aby nás viděli, jak se také učíme, (8) vědění o digitálních technologiích je stejně důležité jako čtení a psaní. Se Seymourem Papertem je spojován i koncept informatického myšlení, který představil ve své knížce *Mindstorms* z roku 1980 Papert jako „mentální dovednost dětí rozvíjenou na základě praktikování programování“. Papert byl přesvědčen, že s počítači mohou pracovat i děti a že prostřednictvím programování se může rozvíjet jejich procedurální myšlení. (Voogt et al., 2015). Papertovi hodně záleželo na tom, aby děti samy se svou přirozenou zvědavostí a nadšením objevovaly podstatu věcí. V používání jazyka spatřoval Papert velký význam pro učení: právě programovací jazyk nabízí dětem mocný nástroj pro komunikaci, pomocí něho může dítě sdělovat počítači své představy, svůj záměr, svůj projekt. „The transaction between the computer and the kid will be some kind of „conversation“ or „questions and answers“ in words and numbers.“ (Papert & Solomon, 1972, s. 9)

Rádi bychom společně s vámi přispěli k naplňování myšlenek Seymoura Paperta, jeho žáků a pokračovatelů. Nabízíme k tomu několik aktivit, projektů, pro práci ve Scratch pro žáky ve věku 12 až 15 let, které předkládáme společně s komentáři a postupy ve formě metodické příručky, která výše uvedené principy a přístupy respektuje.

Metodická příručka **PROGRAMOVÁNÍ VE SCRATCH PRO POKROČILÉ, projekty pro 2. stupeň základní školy**, která se vám dostává do rukou, vznikla v rámci projektu CZ.02.3.68/0.0/0.0/16_036/0005322 OP VVV Podpora rozvíjení informatického myšlení (PRIM) a je určena všem učitelům, kteří chtějí se svými žáky pokračovat v rozvíjení algoritmického a logického myšlení a jejich tvůrčích aktivit v prostředí Scratch. Metodická příručka je určena těm učitelům, kteří chtějí, aby jejich žáci řešili zajímavá témata a problémy a přitom objevovali, proč se počítač chová tak, jak se chová, nebo proč naopak počítač něco neprovede, kde se tedy stala chyba, nebo co se musí udělat, aby fungovala počítačová hra, kterou si žáci vymysleli. Tato publikace je tedy určena všem, kdo chtějí se svými žáky učit počítač realizovat jejich nápady. K tomu, aby žáci mohli počítač „naučit“ něco provádět, musejí s počítačem komunikovat, musejí se (na)učit počítač programovat. Scratch tak plní roli instrumentu v procesu instrumentální zkušenosti (ve smyslu Slavík et al., 2020), během níž si osvojují žáci základní poznatky z informatiky.

Metodická příručka nabízí pro výuku ve škole nebo pro zájmové kroužky 10 různě obtížných a tematicky zaměřených projektů, v průběhu jejichž řešení žáci objevují programovací možnosti prostředí SCRATCH verze 3.0¹, učí se vyjadřovat v jazyce Scratch postupy řešení komplexnějších problémů. Programování těchto projektů volně navazuje na zkušenosti se Scratch, získané například při práci podle učebních materiálů J. Vaníček et al. Programování ve Scratch pro 2. stupeň základní školy (www.imysleni.cz/).

Pokud tato metodická příručka přispěje k tomu, aby se žáci učili přemýšlet o tom, jak naprogramovat svou vlastní hru, nebo jak realizovat svůj vlastní nápad, a pokud žáci pochopí význam základních konceptů (cyklus, proměnná, klonování, seznam, logické operátory, posílání zpráv, podmínky, moje bloky a další) tak, aby je byli schopni používat při programování svých námětů, splnil se záměr autorů této příručky. Projekty zařazené do této metodické příručky mohou žáci doplňovat dalšími efekty, rozvinutím děje příběhu, animováním postav, nakreslením vlastních scén, tvorbou vlastních zvukových nahrávek apod.

Snahou autorů je, aby předložené projekty a hry nebyly jen zajímavou aktivitou pro výuku informatiky zaměřeného předmětu nebo zájmového kroužku, ale aby byly rovněž příležitostí pro uplatnění dovedností a znalostí žáků z dalších předmětů (z matematiky, českého jazyka, výtvarné výchovy, zeměpisu, hudební výchovy, aj.) včetně dovedností používat digitální technologie (např. při úpravě digitální fotografie, při práci s grafickými editory, s digitálním záznamem zvuku a jeho zpracování).

¹ https://en.scratch-wiki.info/wiki/Scratch_Versions

Co by už žáci měli umět, než se pustí do řešení projektů

Autoři předpokládají, že žáci nejsou ve Scratch úplní začátečníci, že už získali se Scratch určité zkušenosti, že už si se Scratch vyzkoušeli činnosti podobné těm, jimž se věnuje učební materiál Vaníček et al. Programování ve Scratch pro 2. stupeň základní školy. Očekáváme konkrétně, že žáci už:

- umějí smazat nepotřebné postavy na scéně, zařadit do projektu novou postavu;
- umějí sestavovat sekvence bloků pro pohyb postavy a její přemísťování na ploše;
- umějí sestavovat sekvence bloků pro kreslení perem, pro změnu kostýmů postavy, pro otisknutí postavy;
- dokážou scénář přečíst, rozebrat, opravit, přeuspořádat, modifikovat;
- si vyzkoušeli práci s předem připravenými vlastními bloky;
- umějí používat cykly (opakuji) při kreslení, umisťovat bloky do bloku opakování;
- se seznámili s blokem pro opakování s podmínkou, tj. dovedou rozpoznat, kdy je podmínka splněna, a kdy ne, umějí rozpoznat, kdy je podmínka testována, dokážou sestavit opakování s podmínkou pro jeho ukončení;
- se seznámili s používáním některých podmínek ze záložky pro VNÍMÁNÍ (např. ověřování podmínky, zda je stisknuta nějaká konkrétní klávesa nebo zda se postava dotýká konkrétní barvy);
- dovedou měnit pozadí (scénu);
- rozumějí principu posílání zpráv;
- umějí pracovat se vzhledem postav (jejich velikostí, efekty, aj.);
- umějí používat bloky pro události.

Autoři rovněž předpokládají, že žáci umí otevřít uložený projekt, že umí svůj projekt uložit na disk a že už se seznámili s tím, jak se kopírují scénáře.

S čím se žáci při řešení projektů při sestavování scénářů ve Scratch setkají

V projektech, zařazených do této příručky, budou žáci poměrně často ve Scratch pracovat s bloky pro:

- | | |
|--------------------------------------|-------------------------------------|
| • změnu kostýmů postav/ pozadí scény | • práci s (logickými) operátory |
| • posílání zpráv | • sestavování nových sekvencí bloků |
| • klonování | • práci se zvuky a melodiemi |
| • vnímání | • práci s perem |
| • práci se seznamy | • práci s událostmi |
| • práci s proměnnými | |

Projekty (Tabulka 1) jsou koncipovány jako učební úlohy, v nichž se objevují různě náročné postupy a koncepty. Některé projekty jsou velmi jednoduché (tzv. *malé projekty*), jiné jsou složitější na porozumění některým pojmům, resp. na programátorské činnosti. Projekty mohou být zařazeny do výuky naprosto samostatně v různém pořadí, obsahově na sebe nenavazují. S výjimkou malých projektů většinou na sebe nenavazují ani z hlediska

osvojovaných dovedností používat Scratch. Projekty se liší také časovou náročností. Není nutné vyřešit se žáky všechny projekty zařazené do příručky, zabralo by to celý školní rok. Bude na vyučujícím, kterým projektům bude ve výuce věnována pozornost.

Než se žáci pustí do řešení některých složitějších projektů, bylo by dobré, aby si nejprve vyzkoušeli některé dílčí postupy ve vybraných malých projektech, které jim usnadní úvahy o tom, jak postupovat při řešení projektu velkého. Konkrétně autoři doporučují dříve, než se žáci pustí do řešení projektu BLUDIŠTĚ, resp. VARIACE NA HRU PIANO TILES, seznámit se s malým projektem NÁKUPNÍ SEZNAM. Obdobně v případě, že se budou žáci věnovat projektu OSTROV POKLADŮ, je dobré vyzkoušet si malý projekt KULIČKA. V těchto malých projektech se žáci připraví na některé nové činnosti se Scratch a naučí programovat dílčí, nicméně velice důležitou část velkého projektu. Malé projekty jsou v podstatě takovými „stavebními kamínky“ velkého projektu, pomocí nichž se žáci seznámí s dílčím „programátorským“ postupem, který pak následně využijí při programování velkého projektu.

Všem projektům je společné sestavování scénářů pomocí bloků prostředí SCRATCH. V praxi to znamená učit žáky přemýšlet o postupech, jejich pořadí a vzájemném propojení, přičemž některé postupy mohou být efektivnější, přehlednější, elegantnější. Důležité je, aby žáci vždycky rozuměli, proč a jaké „pokyny“ počítači zadat, a aby neustále kontrolovali, zda jejich program funguje. Co není v programu, to počítač neprovede. Počítač dokáže udělat jen to, co mu umíme dát za úkol (Keen, 2019). Pokud počítač vykonává něco jiného, než žák očekával, tak je to jen proto, že mu byl takto zadán program.

Výběr projektů

Metodická příručka představuje deset projektů pro práci žáků s prostředím Scratch. Jedná se o programátorské projekty, nikoliv o běžné školní úlohy či o aktivity pro projektovou výuku. Mají-li se žáci něčemu naučit, bylo by vhodné, kdyby se žáci mohli na práci na projektech soustředit, měli dostatek času na přemýšlení o postupech a na to, aby sami zkoumali, jak pokračovat v sestavování programu nebo jak fungují některé bloky Scratch a jejich kombinace. Někdy je vhodnější, aby žáci pracovali samostatně, v jiných případech je vhodnější, aby pracovali ve dvojicích; při práci ve dvojicích (nebo malých skupinách) si mohou žáci vzájemně postup na projektu kontrolovat nebo vysvětlovat, mohou společně diskutovat o zvoleném postupu, mohou se inspirovat jeden od druhého, mohou si vzájemně číst scénáře, analyzovat a interpretovat sestavené sekvence bloků, hledat chyby a zjišťovat příčiny, proč se po spuštění programu neděje to, co očekávali.

Výběr a pořadí projektů

Mezi devíti projektovými aktivitami, určenými žákům 2. stupně ZŠ, resp. žákům nižšího stupně víceletých gymnázií, jsou tři malé projekty a sedm velkých projektů (Tabulka 1):

- *Malé projekty:* Nákupní seznam, Kulička
- *Projekty:* Cestování po světě, Ohňostroj, Interaktivní pohlednice, Ostrov pokladů, Bludiště, Hodiny, Variace na Piano Tiles

Jejich výběr a pořadí pro zařazení do výuky záleží ryze na pedagogickém zvážení učitele. Některé projekty, tzv. malé projekty (Nákupní seznam, Kulička), jsou relativně jednoduché

a časově nenáročné a mohou sloužit jako „rozcvička“ pro práci žáků se Scratch nebo jako příprava na projekty, které už jsou časově i programátorsky náročnější (viz Tabulka 1, v níž je uvedeno, ve kterých projektech lze využít malé projekty).

Rozhodující je, aby se aktivity žákům líbily a aby žáci projekty dokončili tak, aby byly funkční minimálně na některé z úrovní, které jsou specifikovány v metodickém postupu. Žáci by měli prožívat radost, že program funguje a že se jim práce zdařila.

Řešení složitých projektů vyžaduje více času, a to jak na začátku při rozmyšlení a rozhodování, jak se do projektu pustit, kde začít, jak problém strukturovat, jak se vlastně bude projekt řešit, tak při jeho realizaci ve Scratch. K dokončení projektů bude zapotřebí i několik vyučovacích jednotek. Ideální by bylo, kdyby žáci nemuseli svou práci na projektu přerušovat, kdyby mohli na projektu pracovat v delších časových blocích, než je běžná vyučovací hodina (45 minut). Po delším přerušení práce na projektu (např. v důsledku prázdnin, svátků) se může stát, že se žáci budou jen obtížně vracet k rozpracované aktivitě a že jejich zájem o dokončení projektu vyprchá.

Řešení projektů v různé míře přispívá k naplnění očekávaných výstupů učení vymezených pro vzdělávací oblast Informatika a ICT v revidovaných RVP pro 2. stupeň ZŠ (Tabulka 2).

Doporučení, na co klást při práci žáků důraz

1) Ať je všem žákům jasné zadání projektu.

Než se žáci pustí do řešení projektu, přesvědčte se, že žáci vědí, jaký problém budou řešit. Diskutujte s nimi o tom. Nechte je zformulovat zadání úlohy vlastními slovy, aby ukázali, že vědí, co budou dělat, jakou úlohu budou řešit. Během své práce na projektu budou objevovat možnosti Scratch, zkoušet některé bloky, ověřovat své postupy, takže se může stát, že poměrně rychle zapomenou původní zadání úlohy a že se odkloní od původního záměru.

Doporučujeme, aby si žáci zadání řešeného problému vlastními slovy zaznamenali např. slovním zápisem, schématem, tabulkou, mapou, měli je kdykoliv k dispozici, aby se k němu mohli po celou dobu práce na projektu vracet.

2) Ať žáci nejprve přemýšlejí o tom, jak rozložit postup řešení zadaného problému na dílčí části (podúlohy, fáze).

Bylo by dobré, kdybyste dříve, než se žáci pustí do práce, věnovali velkou pozornost tomu, aby žáci začali nejprve přemýšlet o tom, jak zadaný problém rozložit na „menší“ části, s nimiž by si už dokázali poradit, které by na sebe (logicky) navazovaly, postupně se větvily a „zapadaly do sebe jako ozubená kolečka“. Učit žáky rozkládat projekt na podúlohy bude ten největší „pedagogický oříšek“.

Úvahy o tom, jak rozložit problém na dílčí části, můžete vést s celou třídou najednou, nebo se o tom mohou radit žáci ve skupinách, resp. ve dvojicích, nebo rozklad problému na podčásti může každý žák provádět sám. Žáci mohou svoji představu o struktuře rozloženého problému schematicky rozkreslit a představit ji ostatním. Je docela možné, že žáci navrhnou jinou strukturu, než předkládají autoři této příručky. Bude záležet jen a jen na vašem

programátorském a pedagogickém umění, na vašich zkušenostech s prostředím Scratch, na časových možnostech, zda žákům dovolíte, aby strukturovali práci na projektu podle svých představ a postupovali jinak, než je uvedeno v této metodické příručce.

Pokud nebudou podmínky vhodné na to, abyste nechali žáky vymýšlet jejich vlastní členění problému do dílčích částí a poté realizovat ve Scratch, tak pomocnými otázkami navádějte žáky na dílčí kroky řešení (fáze) tak, jak jsou navrženy autory příručky. Fázemi zde rozumíme rozčlenění postupu práce (etapy) na projektu, jejichž výsledkem je funkční program. Toto „fázování práce“ by mělo přispět jednak k tomu, aby si žáci uvědomili, jak lze rozložit problém na dílčí části a jak tyto části budou na sebe navazovat nebo budou propojeny, a jednak k hlubšímu porozumění vybraným blokům a programátorským postupům. Rozložení problému do fází by mohlo žákům pomoci při kontrole funkčnosti řešení.

S rozložením postupu řešení problému na fáze souvisí i to, aby si žáci rozmysleli, kolik (a jakých) postav v projektu bude vystupovat, zda a kdy a které postavy budou viditelné, kde budou umístěny na začátku projektu, jak spolu budou komunikovat, resp. interagovat, zda a jak se budou přemisťovat, kolik scén bude potřeba pro projekt připravit, jak a kdy se budou měnit scény, co bude dělat uživatel (hráč), jak si připraví potřebná data apod. Výsledky těchto úvah mohou žáci zaznamenat pro jednotlivé fáze pomocí tabulky nebo nákresu nebo jako text.

Je zapotřebí počítat s tím, že členění postupu řešení problému do dílčích částí se možná neobejde bez komplikací zejména v případech, pokud žáci dosud nebyli zvyklí tímto způsobem při řešení problémů pracovat. Snažte se, aby každý žák byl při řešení projektu úspěšný minimálně tak, aby jím dosažené řešení projektu bylo funkční, třebaže nedospěl do nejvyšší fáze.

3) Ať žáci věnují pozornost názvům postav, scén, proměnných, seznamů, zpráv, nových bloků.

Ved'te žáky k tomu, aby volili vhodná označení (jména) postav, scén, proměnných, zpráv, seznamů či nových bloků, s nimiž budou ve scénářích projektů pracovat. Zejména v případě zpráv, proměnných, seznamů nebo vlastních bloků by neměli žáci volit názvy nahodile. Dobře zvolené názvy mohou žákům usnadnit orientaci ve scénáři, resp. práci při dalším rozvíjení příběhu projektu nebo jeho modifikaci.

4) Ať žáci neztratí motivaci v pokračování řešení projektu.

Důležité je získat zájem žáků o práci na projektu už od samého začátku. Motivace je klíčovou součástí učení, odvíjí se od zájmů učícího se. Nebojte se se žáky řešit projekt, který se jim zpočátku může zdát obtížný. Pokud se jim podaří projekt vyřešit, budou mít radost, že se jim to povedlo. Dejte jim šanci, aby tuto radost zažili, aby se příště nebáli řešit ještě složitější úlohu. To je v souladu s myšlenkou Seymoura Paperta o hravém učení (angl. playful learning²) a náročné zábavě (angl. hard fun) nebo také s vyjádřením Mitchela Resnicka (2004) „Edutainment? No Thanks. I Prefer Playful Learning“. Práce na projektu by měla být dostatečně zábavná, ale i dostatečně obtížná, přičemž by měla udržet žákovu pozornost a zájem o pokračování v řešení projektu. Učení žáka je závislé na kontextu. Během práce na

² <http://www.papert.org/articles/HardFun.html>

projektu se žák neučí abstraktní izolované poznatky a dovednosti, ale poznává je v souvislostech s příběhem projektu a s řešením konkrétního problému.

K tomu, aby se žákům povedlo ve Scratch naprogramovat funkční projekty, je zapotřebí, aby měli po celou dobu práce na projektu chuť pracovat, poznávat, učit se, nevzdávat se. Budete muset být velice trpěliví a některým žákům i pomáhat, aby je dílčí nezdary neodradily. Snažte se udržet zájem žáků o řešení projektu, a to jak z programátorského hlediska, tak z jeho obsahového a grafického provedení. Někteří žáci se budou více zajímat o obsah, o čem projekt je, jiné žáky bude spíše lákat, jak se projekt naprogramuje. Někteří žáci se budou chtít raději věnovat grafické, resp. zvukové stránce projektu. Někdy nebude jednoduché udržet po celou dobu pozornost žáků a jejich zájem o vyváženost mezi grafickou, obsahovou a programátorskou složkou. V každém případě se snažte, aby žáci měli radost z toho, co dělají, aby je bavilo objevovat nové věci, aby se jim podařilo uskutečnit jejich záměr.

5) Ať žáci objevují, experimentují.

Učení je aktivní proces, který vyžaduje určitý čas, proto se svými žáky nespíchejte s prací na projektech. Dopřejte jim dostatek času na to, aby mohli objevovat, jak se bude program chovat, když se změní některé parametry (hodnoty) nebo když použijí jiný blok (sekvenci bloků) nebo když změní pořadí bloků ve scénáři apod. Ved'te žáky k tomu, aby experimentovali nejen s parametry, ale také s jednotlivými bloky, aby si uvědomili, jak bloky a jejich sekvence fungují.

Podporujte své žáky v tom, aby sami navrhovali, jaký blok (příkaz, podmínky, cykly, proměnné, aj.) v programu použít, jaký nikoliv, aby zkoumali, jak se změní chování postav nebo průběh děje v závislosti na změně různých hodnot nebo při nastavení některých efektů.

6) Ať se žáci učí číst kód.

Občas se žákům stane, že jejich program nebude vykonávat to, co chtěli, jak si představovali. Bude potřeba zjistit, proč se tak děje. Podporujte žáky v tom, aby pozorně napsaný kód prošli, aby si jej (nejlépe nahlas) přečetli a aby při jeho čtení vysvětlili, co se stane a proč. Pokud je to možné, nechte žáky během projektů v některých částech pracovat ve dvojicích, ať společně čtou a interpretují kód, ať si vzájemně kontrolují funkčnost scénáře a vysvětlují, co se bude dít. Jeden od druhého se žáci mohou mnohému naučit.

7) Ať žáci hledají chyby v programu.

Bud'te připraveni na to, že méně zkušené žáky nebo žáky, kteří si vymyslí komplikovaný netradiční postup řešení projektu, budou mít často problémy s tím, že jejich program nebude fungovat, a budou muset hledat chybu v programu. Ved'te žáky k tomu, aby si kód pozorně četli a kontrolovali, zda program dělá to, co má. Dovednost hledat chyby v programu je nesmírně důležitá nejen pro rozvíjení dovedností algoritmicky uvažovat, ale i pro proces učení a pro rozvoj dovednosti učit se.

Seymour Papert spatřoval v chybě velký význam pro proces učení a myšlení. Dejte žákům příležitost dělat chyby a poučit se z nich. Žáky povzbuzujte, aby se nebáli dělat chyby, aby se nevzdávali, když program nebude fungovat tak, jak chtějí. Ať pak zkoumají, proč program

nefunguje a v čem může spočívat chyba. Dejte žákům čas pro jejich vlastní postupy, při nichž mohou chybovat.

8) Ať žáci mají možnost projekt rozšířit, modifikovat, vylepšovat.

Ve většině projektů mohou žáci využívat hotové postavy, pozadí, melodie, zvukové nahrávky, které jsou součástí knihovny prostředí SCRATCH. Tvorba vlastních grafických prvků či zvukových nahrávek může být časově náročná. Zvažte (časové, organizační, technologické, aj.) podmínky a možnosti a poté rozhodněte, zda budou žáci pracovat s hotovými scénami, postavami, grafickými či zvukovými efekty, nebo zda budou tvořit své vlastní. Přesto podporujte žáky v tvorbě či úpravě scén, postav a zvukových efektů; vlastní tvorba či úprava hotových grafických či zvukových prvků může některé žáky nadchnout a vést k originálním výstupům, může jim dodat motivaci pro překonávání problematických částí vytvářeného projektu. Určitě se mezi žáky objeví žáci, kteří se budou snažit při řešení projektu jít svou vlastní cestou. A právě tyto žáci velice často přijdou s netradičním řešením. Možná, že se z těchto řešení poučí nejen spolužáci, ale i vy, učitelé. Nebojte se riskovat a něčemu novému se naučit.

Po skončení každého projektu doporučujeme provést společně se žáky shrnutí toho, čemu se naučili, a navrhovat další úpravy, rozšíření, doplnění, vylepšení projektu.

9) Ať žáci spolupracují.

Pokud to bude vhodné, podporujte spolupráci žáků. Necháváme na zvážení, v jakých případech má spolupráce žáků během řešení projektu význam a v jakých je zase naopak důležité, aby každý žák pracoval sám, resp. ve dvojicích. Při spolupráci, při vysvětlování, proč postupovat zrovna tímto způsobem, při hledání chyb, při diskusi, jak postupovat, se žáci mohou řadě věcí daleko lépe naučit a prožívat radost, že se jim daří problém řešit.

10) Ať se žáci něco naučí.

Pokud žáky práce na projektu zaujme a budou si s problémem „hrát“, je to v pořádku. Podstatné však je klást také důraz na to, aby se žáci něčemu naučili, a to nejen v používání Scratch, ale především v tom, jak problém řešit, jak k němu přistoupit, jak o problému uvažovat, jak postupovat. Bylo by dobré, aby si i žáci uvědomili, že se něčemu naučili a že práce na projektu nespočívala jen v sestavování scénáře z bloků Scratch, ale že museli přemýšlet, hledat cestu „jak na to“ jít.

Jedním z nástrojů, jak zdokumentovat to, že se žák něčemu naučil, může být i to, že žák (v případě skupinové práce žáci) po vyřešení projektu sepíše/sepíší, čemu konkrétně se při řešení projektu naučil (naučili), co pro něj (pro ně) bylo úplně nové. Forma takového záznamu, čemu se žák, resp. žáci naučili, bude na vás. Může mít podobu textu (jakýsi protokol o práci na projektu) nebo komentář k nějaké části scénáře nebo přidání poznámky ve výsledném souboru Scratch (sb3).

11) Ať žáci mají možnost sdílet výsledné projekty a zkušenosti s řešením projektu a prezentovat výsledky své práce.

Podporujte žáky v tom, aby vzájemně sdíleli zajímavá řešení a nápady. Umožněte jim, aby výsledky své práce prezentovali před ostatními, aby vysvětlili, jak postupovali, aby se vyjadřovali k práci ostatních, aby se učili hodnotit práci druhých. Sdílením zkušeností se žáci mohou naučit novým věcem a jiným postupům.

Materiály pro žáky

Ke každému projektu je pro žáky připraven materiál, který žákům slouží jako průvodce při práci na projektu. Pomocí návodných otázek a nápovědy materiál žáky vede k přemýšlení o postupu a k sestavení funkčního programu. Cílem není, aby žáci „opisovali“ hotová řešení a sestavovali program „podle hotového návodu“, ale aby přemýšleli, jak program navrhnout a sestavit, aby v průběhu sestavování scénářů experimentovali a přitom objevovali, jak různé bloky nebo jejich sekvence, parametry a další prvky prostředí Scratch fungují, jak se dají využít k řešení zadaného problému.

Každý materiál je vlastně souborem snímků. Materiál můžete žákům:

- **promítat.** Promítání snímků celé třídy však může narazit na to, že každý žák pracuje svým tempem, že každý žák pro „experimentování“ a „objevování“ potřebuje jinou dobu. Pokud budete jednotlivé snímky žákům promítat, budete všem žákům udávat stejné tempo pro práci na projektu. Někteří žáci pak mohou být netrpěliví, protože už jsou hotovi a chtějí vědět, jak mají pokračovat, požadují, aby už byl promítnut další snímek, jiní žáci budou nespokojení, že ještě nejsou hotovi, že by potřebovali, aby se ještě daný snímek promítal.
- **vytisknout a rozdat v tištěné podobě.** Vytisknuté materiály pak můžete (po určitých etapách) dávat těm žákům, kteří už mají vše hotové a chtějí pokračovat v práci na projektu, nebo naopak těm, kteří potřebují více času na práci a potřebují pracovat svým tempem. Tisk materiálů na barevné tiskárně pro větší počet žáků může být pro některé školy finančně nákladný (tiskárna, papír aj.).

K některým projektům jsou připraveny pracovní listy.

K některým projektům jsou pro žáky připravené ke stažení pracovní soubory vytvořené ve Scratch, s nimiž mohou podle pokynů dále pracovat (doplňovat je, opravovat je, zkoušet s nimi různé činnosti, modifikovat je). To, čemu se při práci s připravenými soubory naučí, pak mohou využít k dokončení velkého projektu.

Materiály pro učitele

Hlavním materiálem pro vás, učitele, je tato příručka, v níž najdete komentáře a metodická doporučení k učebním materiálům pro žáky. U každého projektu se autoři pokusili upozornit na možné zdroje problémů, s nimiž se můžete při práci na projektu se žáky setkat.

Součástí materiálů jsou i soubory s ukázkovými řešeními (jednotlivých fází) projektů ve Scratch. Tyto soubory můžete žákům předvést, aby si dokázali představit, jak by mohl výsledný projekt fungovat.

K některým projektům je připraven pracovní list pro žáky.

Hodně zdaru a zábavy při řešení projektů a programování her se žáky ve Scratch!

Autoři:
Miroslava Černochová,
Petra Vaňková
Jiří Štípek

Tabulka 1 Základní informace o projektech (V závorce jsou u některých projektů uvedeny malé projekty, které mohou žáci využít při práci na projektu.)

Malé projekty	Konstrukty	Základní prvky ze Scratch	Časová náročnost (vyučovací hodiny)	Obtížnost (hvězdičky)	Přesah do dalších předmětů a oblastí i v rámci ICT
NÁKUPNÍ SEZNAM	seznam	proměnná, seznam	20-45 min.	☆	
KULIČKA	nekonečný cyklus	vnímání (klouzej, skoč), zpráva, pero	1	☆	
Projekty	Konstrukty	Základní kategorie ze Scratch	Časová náročnost (vyučovací hodiny)	Obtížnost (hvězdičky)	Přesah do dalších předmětů a oblastí i v rámci ICT
CESTOVÁNÍ PO SVĚTĚ	nekonečný cyklus	změna kostýmu, dotyk s barvou	1-2	☆☆	zeměpis, biologie
OHŇOSTROJ	cyklus, podmínka, paralelní procesy, spuštění události	klonování, grafické efekty, zpráva, změna scény	1-2	☆☆	výtvarná výchova (hudební výchova) ICT (počítačová grafika)
INTERAKTIVNÍ POHLEDNICE	cyklus, podmínka, nekonečný cyklus, spuštění události	dotyk s barvou, zpráva, pero, dotyk s postavou, dotyk s postavou, klonování, proměnná, náhodné číslo, blok, logické operátory, vnímání (stisk klávesy), bublina, otázka, souřadnice	2-3	☆☆☆	výtvarná výchova ICT (počítačová grafika) matematika (hudební výchova)
HODINY	nekonečný cyklus, podmínka, paralelní procesy	pero, zprávy, matematické operace, kostým, souřadnice	2	☆☆☆	matematika (hudební výchova)
OSTROV POKLADŮ (+ Kulička)	cyklus, nekonečný cyklus, podmínka	událost, proměnná, dotyk s barvou, systém souřadnic, bublina, proměnná, změna scény, zpráva, vnímání (ovládání myši)	3-5	☆☆☆☆	ICT (počítačová grafika) výtvarná výchova (zeměpis)
BLUDIŠTĚ (+ Nákupní seznam)	cyklus, nekonečný cyklus, podmínka, paralelní procesy, proměnná	seznam, událost, dotyk s barvou, dotyk s postavou, zprávy, matematické operace	4	☆☆☆☆	matematika
VARIACE NA HRU PIANO TILES (+ Nákupní seznam)	cyklus, nekonečný cyklus, podmínka, paralelní procesy, proměnná	seznam, událost, dotyk s barvou, klon, hudba (tón), logické operátory, změna scény, zpráva	5	☆☆☆☆☆	hudební výchova

Tabulka 2 Naplnění Rámce Očekávaných Výstupů z Informatiky pro Vzdělávací Oblast INFORMATIKA A ICT v činnostech této učebnice (<http://www.nuv.cz/file/4208/>,
<http://www.nuv.cz/file/3361/>) pro 2. stupeň ZŠ

Doména	Žák 2. stupně ZŠ (http://www.nuv.cz/file/3361/)	Programování ve Scratch pro pokročilé. Projekty pro 2. stupeň základní školy. (pro žáky ve věku 13 - 15 let), programování	Označení v textu příručky
1 DATA, INFORMACE A MODELOVÁNÍ	(1) vysvětlí rozdíl mezi daty a informacemi; vyhodnocuje data a informace; odhaluje chyby v cizích interpretacích dat.	projekt VARIACE NA PIANO TILES malý projekt NÁKUPNÍ SEZNAM projekt INTERAKTIVNÍ POHLEDNICE	1-1
	(2) při digitalizaci zvolí formát vhodný pro přenos a uchování informací a svou volbu zdůvodní; v případě potřeby kombinuje data různého typu.	projekt OHŇOSTROJ projekt OSTROV POKLADŮ	1-2
	(3) vymezí problém a určí, jaké informace bude potřebovat k jeho řešení; k popisu používá grafy, případně další ikonické modely.	projekt OSTROV POKLADŮ projekt BLUDIŠTĚ	1-3
	(4) zhodnotí, zda jsou v modelu všechna data potřebná k řešení problému; vyhledá chybu v modelu a ve vlastním modelu chybu opraví; porovná svůj navržený model s jinými modely k řešení stejného problému a vybere vhodnější, svou volbu zdůvodní.	Všechny projekty Projekt HODINY (model otáčivého pohybu hodinových ručiček) Projekt OHŇOSTROJ (model vystřelování hvězdiček) Projekt interaktivní pohlednice (MODEL PADAJÍCÍCH VLOČEK)	1-4
2 ALGORITMIZACE A PROGRAMOVÁNÍ	(5) po přečtení jednotlivých kroků algoritmu nebo programu vysvětlí celý postup; určí problém, který je daným algoritmem řešen.	Všechny projekty Během řešení každého projektu žák musí prokázat, že rozumí jednotlivým krokům algoritmu, programu.	2-5
	(6) rozdělí problém na jednotlivě řešitelné části a navrhne a popíše kroky k jejich řešení.	Všechny projekty U každého projektu se žáci zabývají tím, jak rozdělit projekt na menší celky (tzv. fáze).	2-6
	(7) upraví daný algoritmus pro jiné problémy; ověří správnost postupu navrženého i někým jiným, najde a opraví v něm případnou chybu.	Všechny projekty	2-7
	(8) navrhne různé algoritmy pro řešení problému; vybere z více možností vhodný algoritmus pro řešený problém a svůj výběr zdůvodní.	Všechny projekty Každá fáze projektu předpokládá malý funkční celek, na který navazuje v další fázi. Navrhne algoritmy pro řešení konkrétních zadaných otázek v konkrétní fázi.	2-8

	(9) v blokově orientovaném programovacím jazyce sestaví přehledný program pro vyřešení zadaného problému; program otestuje a opraví v něm případné chyby.	Všechny projekty.	2-9
	(10) používá opakování, větvení programu, proměnné, podprogramy s parametry; používá události k paralelnímu spouštění podprogramů.	Všechny projekty	2-10
3 INFORMAČNÍ SYSTÉMY	(11) vysvětlí účel informačních systémů, které používá, a identifikuje jejich jednotlivé systémové prvky a vztahy mezi nimi.		3-11
	(12) vyhledává, vkládá, upravuje data přes uživatelské rozhraní; řadí a filtruje záznamy v tabulce; využívá při práci s daty v tabulce vzorce a funkce.	projekt BLUDIŠTĚ	3-12
	(13) vymezí problém a určí, zda při jeho řešení využije evidenci dat.	projekt OSTROV POKLADŮ projekt BLUDIŠTĚ	3-13
	(14) nastaví pravidla pro práci se záznamy v evidenci dat.	projekt PIANO TILES (práce se zvuky, tóny a seznamy) projekt BLUDIŠTĚ	3-14
	(15) navrhne a vytvoří tabulku pro evidenci dat.	malý projekt NÁKUPNÍ SEZNAM projekt BLUDIŠTĚ	3-15
	(16) vede navrženou evidenci dat, sleduje dodržování stanovených pravidel a postupů, hodnotí fungování evidence, opraví chyby, případně navrhne vylepšení.	projekt BLUDIŠTĚ	3-16