

MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY



Interaktivní elektronická učebnice informatiky a programování pro žáky I. stupně

DIPLOMOVÁ PRÁCE

Bc. Jakub Špiřík

PODZIM 2015
BRNO

Prohlášení

Prohlašuji, že tato práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Vedoucí práce: MgA. Jana Malíková

Poděkování

Na tomto místě bych rád poděkoval vedoucí své diplomové práce MgA. Janě Malíkové za její ochotný přístup, poskytnuté vědomosti a odborné rady při zpracování tématu. Mé poděkování patří také konzultantovi Ing. Martinu Dosedlovi, Ph.D za cenné rady z didaktického a informatického hlediska. Dále bych chtěl poděkovat všem dětem a jejich rodičům za účast při uživatelském testování. V neposlední řadě děkuji své rodině a přátelům za jejich podporu během období vzniku této práce.

Shrnutí

Práce začíná stručným představením informačních technologií na základních a středních školách, popisuje jejich využití a s ním spojené výhody a nevýhody. Poté následuje krátká kapitola o výuce informatiky na českých školách a přehled látky, kterou by měli zvládat žáci 1. stupně základní školy.

Další kapitola se zabývá návrhem uživatelského prostředí pro děti. Pojednává o schopnostech dětí při práci s počítačem a postupech, které by měly být při designu pro děti zvoleny.

Třetí kapitola je věnovaná výukovým webovým aplikacím a serverům určeným převážně pro děti. Zmíněné webové služby mají jak obecné zaměření, tak i specializaci na výuku informatiky.

Kapitola o použitých technologiích představuje programové nástroje a vybavení, se kterým byla vytvořena aplikace nové učebnice.

V části o tvorbě aplikace učebnice je představena výsledná webová aplikace návrhu učebnice informatiky s názvem Ajtáček - od návrhu, přes strukturu až po detailní popis jednotlivých částí aplikace.

Práce je zakončena kapitolou o testování aplikace dětmi. Zde jsou uvedeny poznatky z odborných zdrojů, samotný popis testování učebnice Ajtáček a také výsledky testování.

Klíčová slova

návrh uživatelského prostředí pro děti, elektronická interaktivní učebnice, webová aplikace, HTML5, Canvas, jQuery, Node.js, uživatelské testování s dětmi

Obsah

Úvod	1
1 Informační technologie ve vzdělávání	2
1.1 Počátky a koncepty IT ve vzdělávání	2
1.1.1 Počítačem podporovaná výuka (Computer-assisted instruction – CAI)	2
1.1.2 Počítačem řízené učení (Computer-managed Learning – CML)	2
1.1.3 Učení podporované počítačem (Computer-assisted Learning – CAL)	2
1.1.4 E-learning	3
1.1.5 Učení podporované webovými stránkami	3
1.2 Zavedení počítačů do škol	3
1.3 Výhody využití IT ve výuce	4
1.4 Nevýhody využití IT ve výuce	4
2 Výuka informatiky na základních školách v České republice	5
2.1 Rámcové vzdělávací programy (RVP) a školní vzdělávací programy (ŠVP)	5
2.2 1. stupně základních škol	6
3 Návrh uživatelského prostředí pro děti	7
3.1 Interaktivní design	7
3.2 Informační grafika a vizualizace pro děti	8
3.3 Positive Technological Development (PTD)	8
3.3.1 Kreativita pro budování sebejistoty	9
3.3.2 Volby chování v rozvíjení charakteru	9
3.4 Rozdíly mezi dětskými a dospělými uživateli	9
3.5 Doporučení pro návrh prostředí v závislosti na vlastnostech dětí na 1. stupni základních škol	10
3.5.1 Fyzický vývoj	10
3.5.2 Kognitivní vývoj	11
3.5.3 Sociální vývoj	12
3.5.4 Koncentrace	12
3.5.5 Zkušenosti	12
3.5.6 Respekt vůči dětem	12
4 Výukové webové kurzy a aplikace	14
4.1 Khan Academy	14
4.2 Moodle	15
4.3 Masivní otevřené online kurzy (MOOC)	16
4.4 Tynker	17
4.5 Hour of Code	18

5 Použité technologie	20
5.1 Node.js	20
5.1.1 Práce s vlákny	20
5.1.2 Architektura řízená událostmi (event-driven architecture)	21
5.1.3 Správa balíků (npm)	22
5.1.4 Porovnání Node.js s PHP	22
5.1.4.1 Paradigma	22
5.1.4.2 Změny kontextu	22
5.1.4.3 Moduly	22
5.1.4.4 Výkon	23
5.2 Framework Express.js	23
5.3 Embedded JavaScript	23
5.4 JSON	23
5.5 jQuery	24
5.6 HTML5	25
5.6.1 Canvas	26
5.7 Adobe Flash Professional CC a HTML5 Canvas	26
5.7.1 CreateJS	29
5.8 Responzivní web design	29
6 Tvorba interaktivní učebnice	31
6.1 Hosting pro Node.js	31
6.1.1 Virtuální privátní servery	31
6.1.2 OpenShift	31
6.1.3 Heroku	31
6.1.4 Roští.cz	32
6.2 Express generator	32
6.3 Směrování (routing)	33
6.4 Struktura aplikace	34
6.5 Učebnice pro základní školy	35
6.6 Grafický návrh aplikace	36
6.7 Webdesign aplikace	38
6.8 Jednotlivé části učebnice	40
6.8.1 Úvodní stránka	40
6.8.2 Píšeme na klávesnici	40
6.8.3 Psací hra	41
6.8.4 Bezpečně s počítačem	42
6.8.5 Kvíz o bezpečnosti s počítačem	43
6.8.6 Programujeme	44
6.8.7 Programovací hra	45
6.8.8 O internetu	48
6.8.9 Kvíz o internetu	48

6.8.10 O šifrách	49
6.8.11 Šifrovací hra	50
7 Uživatelské testování se žáky 1. stupně	52
7.1 Obecná doporučení při uživatelském testování s děmi	52
7.2 Testování interaktivní učebnice Ajtáček	53
7.2.1 Částečné testování	53
7.2.2 Celkové testování	54
7.2.3 Výsledky celkového testování	54
Závěr	56
Použité zdroje	57
Obrazové přílohy	60
Obsah přiloženého DVD	69

Úvod

V dnešní době se informační technologie staly na školách nedílnou součástí výuky. Domnívám se, že ale stále není využitý celý jejich potenciál a především v České republice jsou pomůcky pro výuku informatiky z důvodu rychlého vývoje technologií zastaralé a děti nemají k dispozici podpůrný výukový materiál, který by byl vhodný pro jejich potřeby.

Schopnosti dětí se za poslední desetiletí také velice změnily, právě díky rychlému vývoji nových technologií. S informačními technologiemi se dostávají do kontaktu čím dál mladší děti. Mají k dispozici dotyková zařízení (tablety, chytré telefony apod.) a nebo přímo PC. Tak začínají být počítačově gramotné již v prvních třídách prvního stupně základních škol. Při práci s počítačem a na internetu však na ně čihá stále častěji nebezpečí například ve formě napadení jejich zařízení viry. Dalším problémem je například špatné sezení u počítače, křivení zad apod. Proto je nutné je na to upozorňovat už v raném věku, nepodceňovat jejich schopnosti a komunikovat s nimi na příslušné úrovni.

Hlavním cílem mé diplomové práce je ukázat, jak by takový podpůrný výukový materiál mohl vypadat, a představit technologie, které by k němu mohly být využity. Výsledný návrh byl zpracován po sbírání poznatků a důkladném studiu psychiky a schopností dětí na 1. stupni základních škol, které se dostávají do styku s vyučovacím předmětem o informačních technologích (cca 8 - 11 let). Samozřejmostí bylo hlubší studium poznatků a doporučení týkajících se navrhování uživatelských rozhraní pro děti.

Teoretická část shrnuje poznatky o používání informačních technologií ve vzdělávání, o výuce informatiky na českých základních školách, o schopnostech a psychice dětí a také o návrhu uživatelského rozhraní vytvořeného přímo pro ně. Tuto část zakončuje přehled použitých technologií, kde jsou zahrnuty nové moderní nástroje včetně Node.js nebo HTML5 nebo nové možnosti programu Adobe Flash Professional verze CC.

Praktická část provází samotným vytvářením responzivní webové aplikace elektronické učebnice s názvem Ajtáček (dostupná na adrese <http://ajtacek-0491.rostiapp.cz/>) - od grafického návrhu, přes strukturu až k detailnímu popisu jednotlivých částí, ve kterých se nachází interaktivní informační grafika, velké množství ilustrací, vědomostní kvízy a jednoduché hry. To vše s využitím Node.js a jeho frameworku Express, HTML5, jeho rozhraní Canvas generované programem Flash a také javascriptové knihovny jQuery. Většina didaktického obsahu je převzatá z učebnice „Informatika pro 1. stupeň základní školy“ od Jiřího Vaníčka. Samotná aplikace však není kompletní učebnicí, ale ukázkou možností, jak by taková učebnice mohla vypadat. Součástí práce je také testování výsledné aplikace s dětmi a závěrečné zhodnocení.

Práce předpokládá znalost základů programování především ve smyslu značkovacího jazyka HTML a skriptovacího jazyka JavaScript a také počítá se základními znalostmi ovládání aplikace Adobe Flash.

1 Informační technologie ve vzdělávání

1.1 Počátky a koncepty IT ve vzdělávání

Počátky elektronické výuky a jejích konceptů můžeme nalézt už v šedesátých letech dvacátého století, kdy v podstatě nebyla dostupná žádná multimédia ani počítačové sítě. Jeden z prvních přístupů zahrnoval učení o nových technologiích pomocí těchto technologií. Jako příklad lze uvést programovací jazyky navržené pro děti (například Logo), které si oblíbili nejen žáci, ale také jejich vyučující. Tato výuka dětí útlého věku měla pozitivní dopad na jejich kognitivní vývoj a pomohla také rozvinout jejich schopnosti v matematice a v přírodních vědách. [1, 2]

1.1.1 Počítačem podporovaná výuka (Computer-assisted instruction - CAI)

Začala se také rozvíjet tzv. počítačem podporovaná výuka, kdy žákům pomáhal počítač například při jednoduchých simulacích různých úkolů nebo při drilování. Tato forma fungovala většinou pouze jako doplněk výuky a vyžadovala přítomnost učitele. [1]

Většinou se jedná o dvousměrnou komunikaci mezi počítačem a žákem, kde počítač klade otázky, které nemusí být nutně testovací. Počítač je schopen zpětné vazby a tím pomáhá žákům pochopit zadané téma.

Výuka je silně individualizovaná a počítač v ní přebírá většinu rutinních úkolů učitele.

1.1.2 Počítačem řízené učení (Computer-managed Learning - CML)

Jedním z dalších využití informačních technologií ve vzdělávání bylo dříve v přístupu, který se nazýval počítačem řízené učení. Počítač byl používán k uchování informací o studentovi, o jeho postupu v učení a jeho výsledcích. Tyto informace pak mohl učitel analyzovat a přizpůsobovat tak svou výuku. Využívaly se však většinou s tištěnými materiály. [1]

V rámci tohoto konceptu byly vytvářeny tzv. integrované systémy učení, které předcházely dnešním systémům pro řízení výuky. Dnešní systémy však na rozdíl od těch starších pracují v prostředí počítačových sítí.

1.1.3 Učení podporované počítačem (Computer-assisted Learning - CAL)

Jedním z nejrozšířenějších konceptů moderních technologií ve vzdělávání je učení podporované počítačem – tedy metody, ve kterých jsou počítače používány k učení. Hlavní důraz se zde klade na proces učení žáků. Jedná se spíše o automatizovanou podporu učení pomocí informačních a komunikačních technologií ve vzdělávacích zařízeních. Jako výhody tohoto konceptu lze uvést například rozvíjení dovedností žáků

při řešení problémů a klasifikace těchto problémů, stimulaci pomocí kreativní práce a podporu učení skrze spolupráci, sociální interakci a vyhledávání informací. Počítač zde může hrát roli pedagogicko-psychologicko-diagnostického nástroje, informátora, konzultanta, simulátoru, trenážéru, nástroje výzkumu procesu učení apod. Tento koncept je velmi dobře znám, ale v odborné literatuře a praxi není tolik zmiňován, zčásti i z důvodu rozvoje jednoho z nejnovějších konceptů, a to e-learningu. [1]

1.1.4 E-learning

Tento koncept se dá charakterizovat jako vzdělávací proces, který využívá data v elektronické podobě. Použití IT je podřízené cíli a obsahu výuky a možnostem a může se případ od případu velmi lišit. Široká škála informačních a komunikačních technologií může poskytnout tisíce podob elektronického učení a může být tak použita jak prezenčním, tak i k distančním formám studia. E-learning se snaží, aby se stal běžnou součástí výuky s respektováním svých zákonitostí, jak od studentů, tak i od učitelů. [2]

1.1.5 Učení podporované webovými stránkami

Na e-learning navazuje při rozmachu internetu i učení podporované webovými stránkami. Žáci používají internet k získání informací a zdrojů se zpětnou vazbou učitele nebo řeší přímo úkoly na určitých výukových webech.

Podle Bijana Gillaniho, tvůrce učebních teorií a e-learningových prostředí, mohou webové stránky zastávat role nositele výukového obsahu (učitelé mohou publikovat multimediální učební materiály), komunikačního nástroje, zdroje informací a kreativního nástroje nebo prostředí. [2]

1.2 Zavedení počítačů do škol

S příchodem tzv. informační společnosti přichází produkce a distribuce vědění za pomoci informačních a komunikačních technologií (ICT). Ty pomáhají data a znalosti uchovávat, přenášet a manipulovat s nimi. Čím dál víc se rozrůstá vědění a tím také vzdělávání společně právě s ICT. Z těchto důvodů vystal i požadavek zapojení ICT do procesů výuky do škol. Zpráva OECD Learning To Change: ICT in Schools (2001) uvádí ekonomické důvody (z důvodu potřeb současné a budoucí ekonomiky, významný faktor na trhu práce), sociální důvody (počítačová gramotnost je čím dál častěji využívána a stává přirozeným předpokladem, ICT pomáhají sociálně znevýhodněným i handicapovaným dětem se zapojit do společnosti) a pedagogické důvody (potenciál pro vyučování a učení, proměna vzdělávacích institucí a samotného procesu učení) pro zapojení výpočetních technologií do výuky. [1]

1.3 Výhody využití IT ve výuce

Jednou z hlavních výhod IT ve výuce je zefektivnění práce učitelů, kde jim při jejich práci pomáhají. Počítá se však s tím, že tyto technologie musí učitel se samozřejmostí ovládat a nesmí se domnívat, že počítače převezmou všechnu jejich práci. Dovednosti a počítačová gramotnost učitelů se však každým rokem zlepšují a učitelé si uvědomují, že informační technologie se staly součástí jejich práce. [1]

Učitelé dostávají do rukou mocný nástroj, který jim dokáže pomoci při přípravě výuky a při archivaci těchto příprav i jejich následné aktualizaci.

Žáci nemusí jen konzumovat poznatky, ale díky IT a různým simulacím si mohou danou látku „osahat“ a pracovat s modely přímo ve výuce nebo i mimo ni. Při těchto aktivitách je také podporována komunikace a týmová práce. Při prezentaci výsledků vlastní práce na síti mohou pak získat důkladnější pohled na své prezentační dovednosti a dostává se jim také podpora jejich kreativity nejen v prezentačním odvětví, ale i v samotné práci.

Motivování studentů není jednoduchý proces. Každého jedince může motivovat ke studiu něco jiného. Může to být sama látka nebo její využití. V klasické výuce je výhodou to, že motivovat může samotná osoba vyučujícího, díky tomu, jak učí a jak komunikuje se studenty. V elektronické podobě může být takováto motivace velmi omezená. Je tady ovšem „lákadlo“ v podobě počítače a v podobě počítačového programu, který může být atraktivní nejen vzhledem, ale i formou (například učení ve formě počítačové hry).

Nelze zapomenout na přínos IT pro žáky se speciálními vzdělávacími potřebami. Mohou jim být poskytnuty různé asistivní prostředky, možnosti dálkového vzdělávání. Do vzdělávací komunity se mohou tak připojit i jedinci, kteří bez nových technologií by toho kvůli svému zdravotnímu stavu nebyli schopni. [3]

1.4 Nevýhody využití IT ve výuce

Přestože jsou informační technologie mocným pomocníkem, mohou také snadno představovat nepřehlednou a komplikovanou bariéru a žák nebo učitel mohou nabýt dojmu, že jsou IT ve výuce zbytečné nebo dokonce nebezpečné. Někteří učitelé přistupují k novým technologiím s nedůvěrou, a tím tak vytváří bariéry, které zabraňují plné implementaci informačních technologií do vzdělávání.

Příčiny bariéry mohou vzniknout kvůli nedostatečné dovednosti v práci s IT a nedokonalé počítačové gramotnosti. Pak může přijít na řadu také nedostatek motivace a jistoty v užívání IT. Učitelé odmítají přiznat své omezené dovednosti a odmítají nové technologie používat v hodinách, aby neztratili autoritu u studentů. Za bariéry někdy může také zastaralé nebo nedostatečné pedagogické vzdělání učitelů (učí se pracovat s technologiemi, ale na didaktickou práci s těmito zařízeními nejsou připraveni). Za nechutí k IT je někdy také nedostatek možností nebo času věnovat se rozvíjení dovedností v IT nebo také nedostatek kurzů a workshopů, které by pomohly vyučujícím při jejich implementaci informačních technologií do výuky. [3]

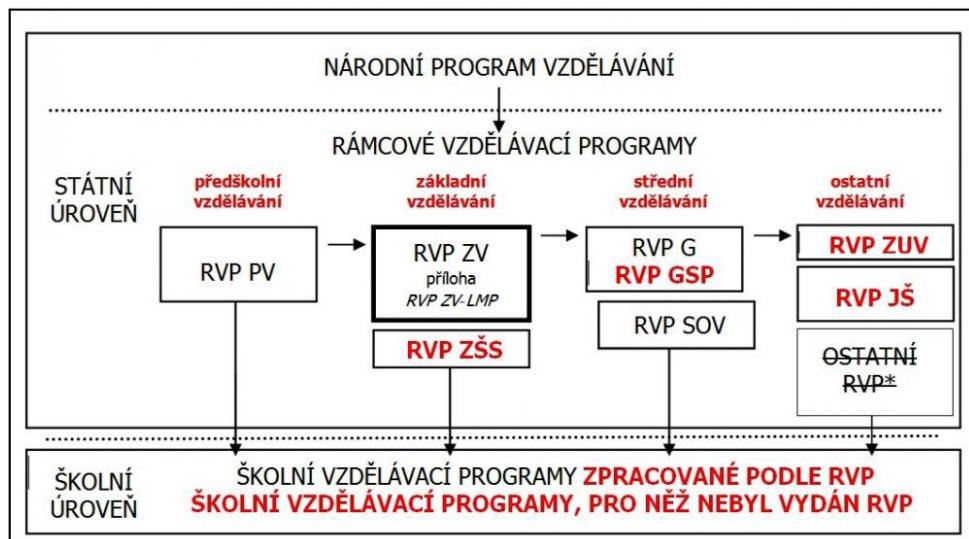
2 Výuka informatiky na základních školách v České republice

V dnešní době existují informační technologie všude kolem nás a používá je drtivá většina zaměstnání. Znalost práce s počítačem a jeho prostředky je tedy důležitá pro budoucí uplatnění na trhu práce.

Děti většinou znají počítač už z domova a setkávají se s ním již od útlého věku. Berou ho však převážně jako nástroj určený pro zábavu. Důležité tedy je ukázat jim funkci počítače i jako mocné pracovní pomůcky, která jim může usnadnit nebo zefektivnit práci.

2.1 Rámcové vzdělávací programy (RVP) a školní vzdělávací programy (ŠVP)

Rámcově vzdělávací program se v České republice používá jako nejvyšší dokument pro vzdělávání společně s projektem Národní program pro rozvoj vzdělávání (tzv. Bílá Kniha). RVP definují povinné rámce pro předškolní, základní a střední vzdělávání. Na jednotlivých školách se pak realizuje výuka podle ŠVP, které jsou založeny právě na RVP. [4]



Obrázek 2.1: Systém kurikulárních dokumentů. [4]

Po schválení zákona o předškolním, základním, středním, vyšším odborném a jiném vzdělávání (dále školský zákon) a Rámcového vzdělávacího programu pro základní vzdělávání ze dne 24. 8. 2004 už učitelé nejsou nuceni připravovat výuku podle

tradičních a zaběhnutých osnov, ale mohou ji více přizpůsobit vlastním zkušenostem a s ohledem na potřeby žáků. Pro tvorbu ŠVP existuje od Ministerstva školství, mládeže a tělovýchovy dokument s názvem Manuál pro tvorbu ŠVP. Tento text popisuje, jak by měl program vypadat a co by měl obsahovat. V ŠVP škola neuvádí, co všechno žáky naučí, ale co budou studenti po absolvování daného ročníku umět. [4]

2.2 1. stupně základních škol

Podle současné školní reformy a rámcových vzdělávacích programů pro základní vzdělávání byla výuka informačních a komunikačních technologií zařazena jako povinná součást na 1. a 2. stupni. Program také prezentuje základní téma výuky a jejich výstupy. Na 1. stupni by se žáci měli seznámit se základy práce s počítačem, poučit se o vyhledávání informací a komunikaci a tyto informace využít a zpracovat. Na 2. stupni by se pak učivo o získávání a zpracování informací mělo prohlubovat. Žáci by měli být také poučeni o hygieně práce s počítačem a také o etice na internetu. [5]

Na 1. stupni by žáci měli dosáhnout základní počítačové gramotnosti, pokud jí již nedosáhli v předškolním věku. Důraz je kladen na práci s informacemi, na jejich tok, vznik, přenos, zpracování a vyhledávání. Součástí plánu je také zvládnutí základních způsobů komunikace v IT a základních softwarových pomůcek (grafický editor a textový editor) a také získání schopnosti algoritmického myšlení při práci s počítačem. [5]

Určování věrohodnosti informace a její zpracování nebo využití je v plánu zařazeno až na 2. stupeň. Děti by však měly zpochybňovat a alespoň na základní úrovni ověřovat informace, které najdou na internetu, k němuž mají čím dál snazší přístup.

3 Návrh uživatelského prostředí pro děti

Na základních školách se vnímání a schopnosti dětí rapidně rozšiřují a zlepšují, ale můžou vznikat velké rozdíly mezi jedinci stejného věku. Různě se vyvíjejí jejich kognitivní schopnosti během jejich působení na základních školách, kdy získávají základní znalosti z odlišných oborů. Žáci jsou závislí na kvalitní zpětné vazbě a sociální interakci. Jsou si také dobře vědomi svého vztahu ke „světu dospělých“.

Při komunikaci s počítačem cítí děti většinou nechuť číst jim neznámé nebo nové informace a raději využívají reakce uživatelského rozhraní na jejich činnost.

Dnes však v životě dětí hrají počítače daleko větší roli a do těchto technologií se také vkládá mnohem více důvěry. Je proto důležitý rozvoj povahových vlastností, které dětem zaručí bezpečný způsob, jak se spojit a komunikovat s ostatními. Stejně tak je důležité dávat příležitosti zlepšovat jejich dětský svět skrze jejich technologické dovednosti a dovést je k novým způsobům myšlení.

Všechny elementy rozhraní by měly být navrhovány v globální perspektivě aplikace. Každý aspekt rozhraní musí být brán z pohledu celé aplikace a ne jen jako individuální prvek ovládání nebo vizuální element. Uživatelova zkušenosť s aplikací by měla reprezentovat harmonii mezi formou, obsahem a funkcí. [6]

Čistě estetické prvky by neměly do rozhraní zasahovat s výjimkou právě u vzdělávacích aplikací nebo her, zvláště těch pro děti, kde je místo pro větší experimentování. Vizuální zkušenosť z rozhraní a obsahu slouží totiž jako velká část zábavy. Porušení tohoto doporučení může posílit vztah mezi obsahem a ovládáním. I v těchto případech by se mělo zacházet s vizuální stránkou rozhraní s rozvahou, aby se uživatel v obsahu vyznal a aby nevznikl vizuální šum, který rozptyluje uživatele a brání mu ve snadném dosažení hlavního cíle. [6]

3.1 Interaktivní design

V uživatelském prostředí jsou pro děti velmi atraktivní interaktivní prvky. Ty však mají také svá pravidla. Pojem interaktivního designu definuje obor studia, který se zaměřuje na smysluplnou komunikaci médií skrze kolaborativní procesy mezi lidmi a technologií. Správný interaktivní design je jednoduchý, srozumitelný s jasnými cíli a intuitivním rozhraním. [7]

Interaktivní design se nesmí plést s pojmem interakční design. Interaktivní design se zaměřuje na vytváření smysluplných využití systémů a hardwaru, který navrhuje právě interakční design.

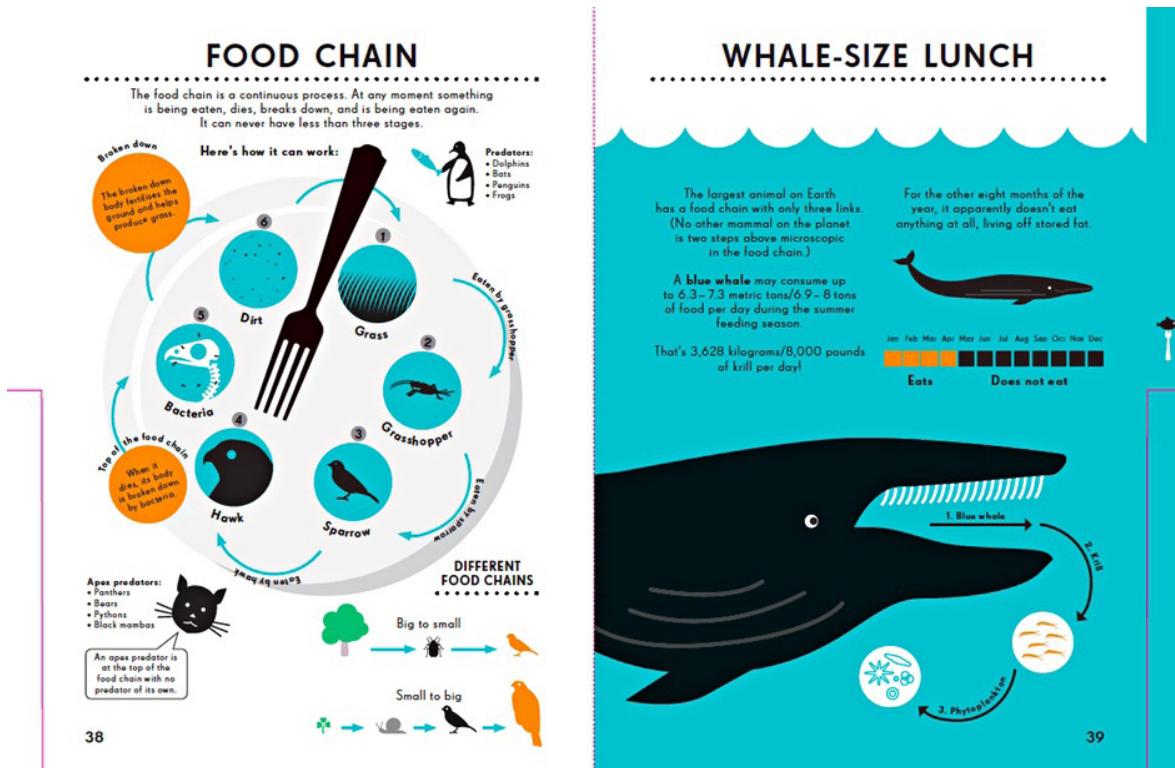
Důležitým aspektem je uvědomit si, že při vytváření interaktivních artefaktů není hlavní estetika, nejdůležitější je pochopit koncového uživatele a splnit jeho potřeby nebo očekávání. Všechny elementy designu aplikace nebo webu by měly být intuitivní

a mělo by být jasné, že se s nimi dá interagovat. V ideálním případě by měl uživatel předpokládat i to, co se stane, když přejede nebo klikne na daný prvek. [8]

3.2 Informační grafika a vizualizace pro děti

Prezentování dat pouhým textem může být pro děti velmi nudné a většinou k tomuto způsobu mají odpor. Děti mají pokročilý vizuální jazyk a jsou schopné lépe vnímat pomocí obrázků, je tedy možné ve vizualizacích více experimentovat. V informačních grafikách by měl mít obrázek stejnou infomační „sílu“ jako text a data by měla být podávána spíše jako příběh než suchý dekorovaný výčet. Psychologické studie ukazují, že je efektivnější dítěti něco přímo ukázat než mu to pouze sdělit. [9]

Pro děti mohou být reálná data, pokud jsou správně prezentována, velice lákavá v kontrastu s fantazií, se kterou se setkávají např. v herním průmyslu nebo v knihách pro děti. Určitá fakta můžou být pro ně více šokující než fakta z některého z imaginárních světů.



Obrázek 3.1: Příklady informačních grafik pro děti. [9]

3.3 Positive Technological Development (PTD)

Vytváření a přetváření digitálních světů pro děti a mladistvé, kde by si mohli hrát, interagovat a hlavně se učit, je velmi aktuální téma. Návrhy virtuálních prostředí jsou

zatím motivovány více komerčními potřebami než touhou po způsobu, jak pozitivně ovlivnit vývoj našich dětí. Pro vývoj takového digitálního prostředí je vhodné se řídit nějakými zadanými pravidly, které budou částečně kontrolovat naše rozhodnutí. Před více než deseti lety vymysleli vědci v MIT Media Lab a v Tufts framework sadu doporučení, které dali název Positive Technological Development (Pozitivní technologický rozvoj). Následující vybrané koncepty spadají právě do PTD. [10]

3.3.1 Kreativita pro budování sebejistoty

Nové technologie dovolují dětem vymýšlet nové nápady a vyjádřit je obrovským množstvím způsobů, jaké např. před deseti lety nebyly možné. Když děti vytváří nějaký svůj projekt, který je založen na používání počítače, musí překonávat i různé technické problémy. Vytváří si tak důvěru ve svůj učební potenciál a rychleji se učí ovládat nová zařízení, a často je používat novým kreativním způsobem. [10]

3.3.2 Volby chování v rozvíjení charakteru

Dětství a puberta jsou období života, ve kterém se člověk vyvíjí jako samostatný morální jedinec. To zahrnuje míru erudovanosti dělat informovaná rozhodnutí s vyhodnocováním možných důsledků. Nové technologie poskytují úžasné „hřiště“ ve virtuálních světech pro experimentování s různými možnostmi a řešení otázek typu „kdyby“. Je zde samozřejmě pořád i možnost riskování, protože mladí uživatelé nemusí být upozorněni na všechny možnosti vývoje situace předem. Poté mohou zkoumat svoje rozhodnutí a posoudit své charakterové rysy. Díky tomuto způsobu si mohou děti vytvořit morální cit a jednat spravedlivěji a zodpovědněji. [10]

3.4 Rozdíly mezi dětskými a dospělými uživateli

Většina aspektů použitelnosti se za posledních deset let nezměnila. Například se nemění doporučení (guidelines), jak nejlépe strukturovat menu a kolik položek by mělo obsahovat. Tato doporučení jsou totiž více závislá na charakteristikách a omezeních lidského mozku než na vývoji technologií.

Výjimku tvoří například přístup k videím na webu (dříve skepticismus a dlouhé časy načítání, dnes běžná praxe a oblíbená činnost většiny uživatelů internetu), který se za poslední dekádu změnil díky zlepšení počítačových sítí, které dokáží přenášet stále větší rychlostí.

Zkušenosti dětí s výpočetními technologiemi se však rapidně změnily tím, že jejich čas strávený u počítače se průměrně ztrojnásobil (podle výzkumu Kaiser Family Foundation). Podle Nielsen Norman Group je pro zjištění, jak děti ovládají práci s webovými stránkami, důležitá jejich praxe a čas, který stráví „online“. [11]

	Děti	Dospělí
Cíl při prohlédávání webu	zábava	komunikace, udělat úkon/práci
První reakce	velký důraz na první pohled/ pocit ze stránky	velký důraz na první pohled/ pocit ze stránky
Ochota čekat	požadována okamžitá reakce	limitovaná trpělivost
Chování při prozkoumávání	zkoušení mnoha možností, prohledávání obrazovky	drží se hlavních cest pro splnění úkolu
Redundantní navigace	velice matoucí	mírně matoucí
Animace a zvuky	vítané	většinou odporné
Velikost fontu	12 bodů (pro starší děti)	10 bodů (14 pro seniory)
Inzeráty a reklamy	většinou nedokáží rozeznat od reálného obsahu	inzerátům se vyhýbají nebo k nim přistupují skepticky
Vyhledávání	větší závislost na záložkách než vyhledávání	hlavní přístup k webu

Tabulka 3.1: Příklady rozdílných reakcí dětských a dospělých uživatelů. [11]

Děti stráví mnohem více času přizpůsobováním vzhledu jejich virtuálního prostředí, pokud je jim tato možnost dána. Jsou zvědavé a neváhají zkoušet nové věci tím, že budou klikat na všechno, co uvidí na monitoru (na rozdíl od dospělých jedinců, kteří mají zkušenosti s danou technologií nebo jsou poučeni o její bezpečnosti). [12]

3.5 Doporučení pro návrh prostředí v závislosti na vlastnostech dětí na 1. stupni základních škol

3.5.1 Fyzický vývoj

U dětí nemůžeme díky stále probíhajícímu vývoji jejich těla počítat například s průměrnou velikostí ruky a prstů, kterými budou ovládat zařízení. Jejich jemná motorika také nemusí být plně vyvinutá a mohou mít např. problémy poklepávat na menší tlačítko.

Až ve věku 10–12 let se stávají fyzicky dospělými. Bez rozdílu si však stále rády hrají, ale přesto samy sebe mohou pokládat už za dospělé. Mají také potřebu být důležitým a užitečným článkem v nějaké větší skupině. Svoje nápady a vize začínají konfrontovat s dospělými. [13]

Děti se také zapojují do týmových sportů. Mají tedy rády grafiky, které implikují týmového ducha, i když uživatel (dítě) daný sport aktivně neprovozuje. [13]

Ve vyšších třídách se velmi vyvíjí jejich koordinace oko-ruka a dokáží ocenit detailní grafiky a animace. Jsou také již zběhlí v počítačových hrách.

Pro navýšení aktivity potřebují více odpočinku. Mohou být pod stresem. Proto je dobré vytvářet design, který umožňuje přestávky (k tomuto efektu mohou pomoci studené a klidné barvy). Aplikace, které mohou využít i v klidu o samotě, jsou u dětí velmi oblíbené. [13]

Čtení na displeji nepůsobí dětem žádné dlouhodobé poškození zraku. Jde o stejný proces pro oči jako u čtení z papíru. Pouze u čtenářů nižšího věku, kteří se teprve učí číst a většinou jsou k textu fyzicky blíže bez ohledu na velikost fontu, hrozí únava, rozostřené vidění a další problémy, pokud mezi čtením nebudou dodržovat přestávky. Toto platí opět jak pro displeje, tak i pro tištěná média. [14]

3.5.2 Kognitivní vývoj

U dospělých můžeme předpokládat, že jejich mentální schopnosti jsou podobné a jsou zvyklí komunikovat a interagovat přirozeným způsobem. Například je možné jim zadat úkol a pokud budou informace v zadání nedostatečné, položí dotaz příslušné autoritě nebo vyhledají sami řešení. U dětí s tímto systém nelze vždy počítat a je nutné tomu přizpůsobit uživatelské rozhraní a také testování. [15]

Obecně také platí, že čím menší dítě, tím je víc egocentrické – myslí si, že všichni ostatní vidí svět stejně jako ono. Jak rostou, začínají vidět svět i z jiné perspektivy. [15]

Dozvídají se více o světě, včetně mýtů a biografií – mohou se zajímat o to, jak věci pracují. Přisvojují si také nové matematické schopnosti, dokáží řešit úkoly s manipulací objektů v prostoru nebo s jejich počtem. Jsou také schopny pracovat pod časovým stresem nebo hospodařit s penězi na základní úrovni. Získávají povědomí o světě a přírodních zákonech, a o tom, co je způsobuje. Tím u nich roste zájem o i přírodní vědy. Děti začínají rozumět také hříčkám a vtipům. [13]

Čím je jedinec starší, tím bývá i zdatnějším čtenářem. Jednoduché instrukce a delší bloky textu jsou tak většinou dětmi tolerovány. Stále je ale dobré k textu připojovat hodně obrázků.

Děti mohou začínat pracovat na delších a komplexnějších projektech, kde je jejich zadání přesně formulováno – mohou být vhodné složitější grafické prvky. Návrhy, které dětem mohou pomoci s organizováním nebo rozčlenováním plochy návrhu, jsou tedy přínosné.

V pozdějším věku zvládnou myslet několik kroků dopředu – strategické hry, hádanky, dlouhotrvající úkoly jsou pro ně již lépe zvládnutelné. [13]

3.5.3 Sociální vývoj

Každý člověk se na začátku života teprve učí sociálnímu cítění a hodnotám. Většina lidí také dokáže být empatická a držet se pravidla „chovám se k ostatním tak, jak bych chtěl, aby se oni chovali ke mně.“

Děti v tomto případě mohou být velice sobecké. Velmi často se stává, že upřednostňují svoje přání a potřeby nad potřebami ostatních. Sociální vývoj je samozřejmě u každého jiný. Záleží například na výchově, charakterových rysech a osobních preferencích. [15]

Vzrůstá důležitost přátelství – děti mají rády grafiky, které vyjadřují, že jsou součástí nějaké větší skupiny. Design, který zahrnuje symbolický nebo kódovaný jazyk, který je přístupný této skupině, je u dětí velmi oblíbený.

Čím dál častěji se setkávají s novými domněnkami a názory, a tak barvy, textury a vzory, které jsou odvážné nebo experimentální, mohou být pro ně více fasinující.

3.5.4 Koncentrace

Narozdíl od dospělých jedinců se děti dokáží snadno nadchnout. Nemají tak dobrý odhad, jak dlouho by zadaný úkol mohl trvat a jaké úsilí musí vynaložit. Je tedy velice důležité nadšení u nich udržet co nejdéle pomocí dobré motivace, protože děti jsou velmi spontánní, snadno ztratí pozornost a koncentraci. [15]

Při dlouhém načítání části aplikace existuje možnost zabavit děti nějakou „mini-hrou“ (například několik obrazců, které budou vydávat různé zvuky). Tato forma není však vhodná pro důležité a hlavní aspekty designu webové stránky. Může tím nastat negativní rozptýlení, zhoršená navigace na stránce a ztráta koncentrace.

3.5.5 Zkušenosti

Jeden z nejzásadnějších handicapů oproti dospělým je nedostatek životních zkušeností. Dospělí mohou čerpat inspiraci k řešení současného problému z mnoha více zážitků a situací. Pro dospělé je možné také navrhnut rozhraní, které může připomínat rozhraní například velmi rozšířených kancelářských aplikací, v nichž každý počítačově gramotný člověk se dokáže rychle zorientovat. U dětí s tímto principem počítat nemůžeme. Nejsou ani tak často zvyklé aplikovat známé koncepty na nové situace. Je vždy nutné poskytnout dítěti co nejvíce informací a nepředpokládat, že mají nějakou znalost. Při špatné informovanosti může dítě lehce ztratit pozornost i zájem. [15]

3.5.6 Respekt vůči dětem

Děti jsou dnes již od útlého věku sofistikovanými konzumenty. Jsou citlivé na obsah a design určený určitému věku uživatelů a je důležité nepodceňovat jejich schop-

nosti. Nejlepší metodou při tvorbě produktu je ukazovat a nechat si okomentovat návrhy členy dané cílové skupiny, protože pro dospělého člověka je nemožné se vcítit do myšlení dětí. Prožili totiž dětství v jiné době s jinými technologiemi. [16]

4 Výukové webové kurzy a aplikace

4.1 Khan Academy

V roce 2006 založil absolvent MIT a Harvardu Salman Khan neziskovou organizaci Khan Academy. Ta poskytuje kvalitní vzdělání pomocí výukových videí a interaktivních cvičení dostupná odkudkoliv s připojením k internetu. Je to také první bezplatná virtuální škola na světové úrovni s nepřeberným množstvím obsahu z mnoha různých oborů.

Khan na myšlenku videolekcí přišel, když doučoval svou sestřenici matematiku. Tato a další videa pak nahrával pro své přátele a příbuzné na server YouTube, kde získala velkou popularitu studentů z celého světa. To přimělo Khana opustit svou bývalou práci a věnovat se naplno tvorbě výukových videí.



Obrázek 4.1: Značka Khan Academy. [17]

Ve svých videolekcích Khan píše a maluje tabletem a při tom je nahráváno jeho mluvené slovo, které popisuje a vysvětluje každou aktivitu na obrazovce. Khan se chtěl od začátku vyhnout myšlence „učitele u tabule“ a preferuje řešení, kdy učitel vysvětluje řešení příkladu přímo jakoby vedle studenta na list papíru. Snaží se překonat vžitý model školní výuky díky využití softwaru, který tvoří testy, známkuje a hlídá jednotlivé studenty a také analyzuje jejich výsledky. Přidělený učitel nebo rodič pak může tyto statistiky lehce kontrolovat. Další výhodou oproti klasické výuce je, že student si může video pozastavit nebo přehrát určitou pasáž znova. [17]

Při interaktivních cvičeních je brán ohled na studentovy nasbírané zkušenosti v kurzech (pokud si uživatel sám nenastaví jinak) a jednotlivé problémy jsou generovány náhodně, tzn. nikdy nedojdou podklady pro praktická cvičení k danému tématu. Jako motivace jsou zde uplatňovány herní mechanismy v podobě odznáčků, které studující dostává za splnění určitého oddílu nebo určitého počtu kurzů.

V budoucnu je plánovaný systém pro doučování mezi spolužáky podle nasbíraných dat o studentech.

$$\begin{bmatrix} 1 & -3 \\ 7 & 5 \end{bmatrix} \times \begin{bmatrix} 10 \\ 12 \end{bmatrix} = \begin{bmatrix} 2 \cdot 10 + -3 \cdot 12 \\ 7 \cdot 10 + 5 \cdot 12 \end{bmatrix} = \begin{bmatrix} -16 & 10 \\ 130 & -60 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 5 \\ 7 \end{bmatrix} = \begin{bmatrix} 1 \cdot 5 + 2 \cdot 7 & 1 \cdot 6 + 2 \cdot 9 \\ 3 \cdot 5 \end{bmatrix}$$

Obrázek 4.2: Snímek z videolekce o počítání s maticemi. [17]

Khan Academy získala celosvětový ohlas a popularitu, a tak není divu, že vznikl zájem o jazykové mutace jednotlivých kurzů. V České republice bylo v roce 2012 založeno sdružení Khanova škola, které má za cíl právě překlad této oblíbené virtuální školy do češtiny. Do roku 2015 bylo přeloženo přes 2700 lekcí a do českého jazyka předabováno přes 110 videí. Tyto překladatelské aktivity podporují společnosti Scio, Google i Microsoft. [18]

4.2 Moodle

Moodle je bezplatný online výukový systém poskytující svým uživatelům možnost snadno vytvořit a spravovat svou vlastní webovou stránku s dynamickými vzdělávacími kurzy, které mohou sloužit jako podpůrné výukové materiály nebo i jako elektronický kurz.



Obrázek 4.3: Značka projektu Moodle. [19]

Autorem Moodlu je Australan Martin Dougiamas, který se online výuce věnoval už za svých studií na technologické univerzitě Curtin, kde také na toto téma napsal diplomovou práci. První prototypy systému Moodle byly napsány v programovacím jazyce Python, pozdější verze byly přepsány do jazyka PHP. [19]

Samotný název Moodle je akronym pro Modulárně objektově orientované dynamické prostředí pro výuku (Modular Object-Oriented Dynamic Learning Environment). Místo slova „Modular“ se používalo i „Martin's“ podle autora projektu. Pro zvolení tohoto názvu hrála také roli skutečnost, že doménové jméno Moodle bylo volné.

Návrh a vývoj systému Moodle je řízen pomocí sociálně konstruktivistického přístupu ke vzdělávání. „Sociální konstruktivismus zdůrazňuje nezastupitelnou roli sociální interakce a kultury v konstrukci poznatků.“¹ Učení je proces nejen osobní, ale i sociální. Jednotliví studenti spolupracují na budování společných poznatků a porozumění významů.

Při vytváření kurzu má tvůrce kurzu možnost využít řadu modulů, z kterých vytváří obsah. Jednotlivé moduly lze dále nastavovat nebo upravovat dle vlastních potřeb. Nové moduly může s pomocí PHP kdokoli vyvíjet a tvořit (to také přispělo k masovému rozvoji systému Moodle). Je možné vkládat HTML stránky, soubory ke stažení, animace, diskusní fóra, úkoly pro účastníky, automaticky vyhodnocované testy, databáze a mnoho dalších technologií. Učitel může také snadno sledovat pokrok svých studentů pomocí statistik, které systém generuje. Do Moodlu lze integrovat i celou řadu dalších systémů např. autentizační nebo komunikační, nebo integrovat systému pro správu obsahu (např. Joomla nebo Drupal).

Tvůrci kurzů ale nemají takové možnosti a často ani schopnosti vyprodukovať kvalitně graficky zpracovaný učební materiál a systém Moodle jim většinou neposkytuje žádné nástroje. Vzniká tak mnoho graficky nekonzistentních materiálů, ze kterých děti čerpají informace velmi těžko.

4.3 Masivní otevřené online kurzy (MOOC)

MOOC (z anglického massive open online courses) je zkratkou pro vzdělávací online (přistupuje se k nim pomocí webového rozhraní) kurzy s neomezenou kapacitou. K tradičním výukovým materiálům jako videa, texty a testy patří v MOOC také interaktivní uživatelská fóra, která pomahají vybudovat komunitu pro studenty, vyučující a jejich asistenty. K nejznámějším a nejpoužívanějším portálům patří například Coursera, Udacity nebo edX. Většina kurzů se v dnešní době spíše zabývá technickými obory.

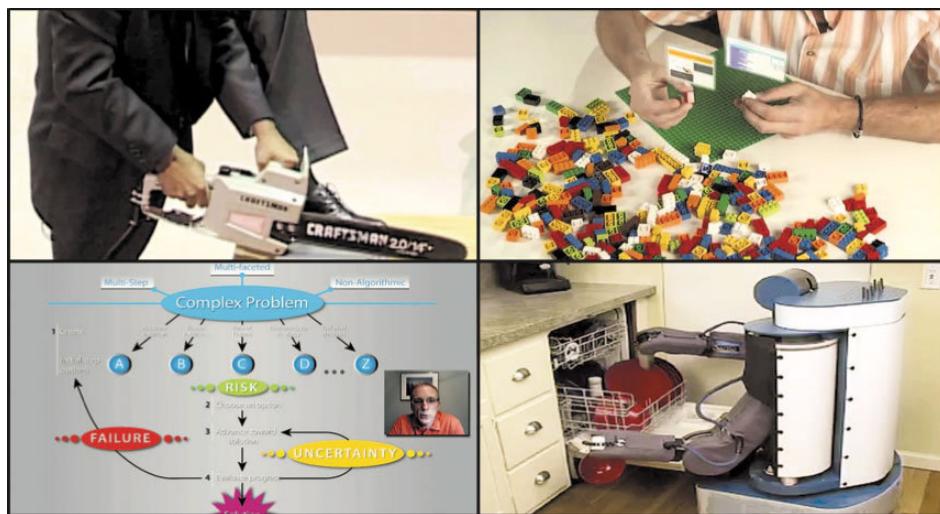
Tyto kurzy jsou výsledkem vývoje v dálkovém studiu, který se objevil v roce 2012. První MOOC si osvojily principy konktivismu a otevřených licencí pro obsah, strukturu a výukové cíle při znovupoužívání a inovování studijních zdrojů. Některé pozdější kurzy mají povahu uzavřených licencí. Svým studentům však poskytují přístup ke studijním materiálům zdarma. [21]

Hlavním médiem pro výuku jsou klasicky přednášky. Videa však můžou být v průběhu zastavována a v přestávkách musí student vyplnit test, který ověří, zda správně pochopil učivo. V kurzech programování je test vystřídán úkolem, kde testování musí napsat kód k vyřešení úlohy. Zpětná vazba je poté elektronická kvůli velkému

¹ [20] s. 3

počtu studentů. Zpětná vazba od reálného člověka se odehrává až na diskusních fórech, která monitorují vyučující nebo jejich pomocníci. Součástí kurzu mohou být domácí úkoly a na konci kurzu se může objevit i závěrečný test a nebo závěrečná práce (podobně jako u prezenčních kurzů na univerzitách).

S MOOC však přichází i různé problémy. Většina zájemců o kurzy neodhadne buďto složitost kurzu nebo své časové možnosti a daný kurz nedokončí. Pokud je kurz zdarma, nemají ani tak motivace ho dokončit. Dalším problémem je známkování, které může být zvláště u netechnických předmětů problematické. Běžnou realitou při vypracovávání domácích úkolů je také podvádění. Tento problém řeší Udacity a edX sledovanými testy. Přesto projde obrovské množství studentů kurzy úspěšně bez podvádění.



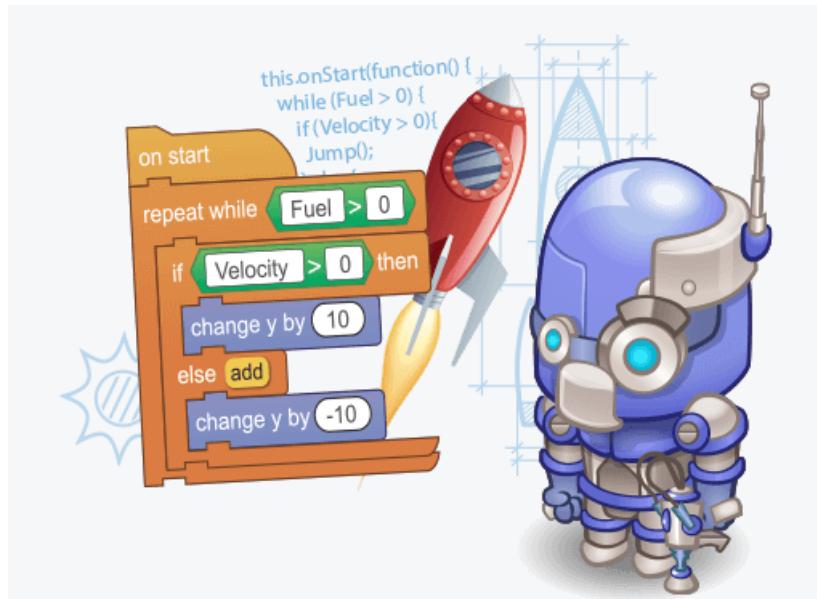
Obrázek 4.4: Ukázky z videolekcí různých MOOC. [22]

Podle Raye Schroedera, ředitele Centra pro online vzdělávání, výzkum a služby na univerzitě v Illinois, jsou pro online vzdělávání ze všeho nejdůležitější kvalitně zpracované materiály, zapojení učitele a interakce mezi studenty. Materiály jsou díky zapojení vysokoškolských pedagogů většinou na velmi dobré úrovni. Protože však se těchto kurzů účastní velký počet studentů, není zapojení učitele jednoduché. [22]

Většina těchto kurzů je však v angličtině a jsou ve většině určené vysokoškolským studentům nebo pracujícím. Alternativy pro děti zatím nejsou tak rozšířené.

4.4 Tynker

Tynker je kreativní počítačová platforma, kde se miliony dětí učí programovat a vytvářet hry a aplikace. Nabízí online kurzy, kde děti mohou postupovat vlastním tempem a učit se doma. Nabízí však také zábavné osnovy pro školy. Za cíl si dává poskytnout dětem pevné základy přírodních věd, programování a kritického myšlení. Bohužel tato platforma existuje pouze v angličtině.



Obrázek 4.5: Ukázka vizuálního programovacího jazyka aplikace Tynker. [23]

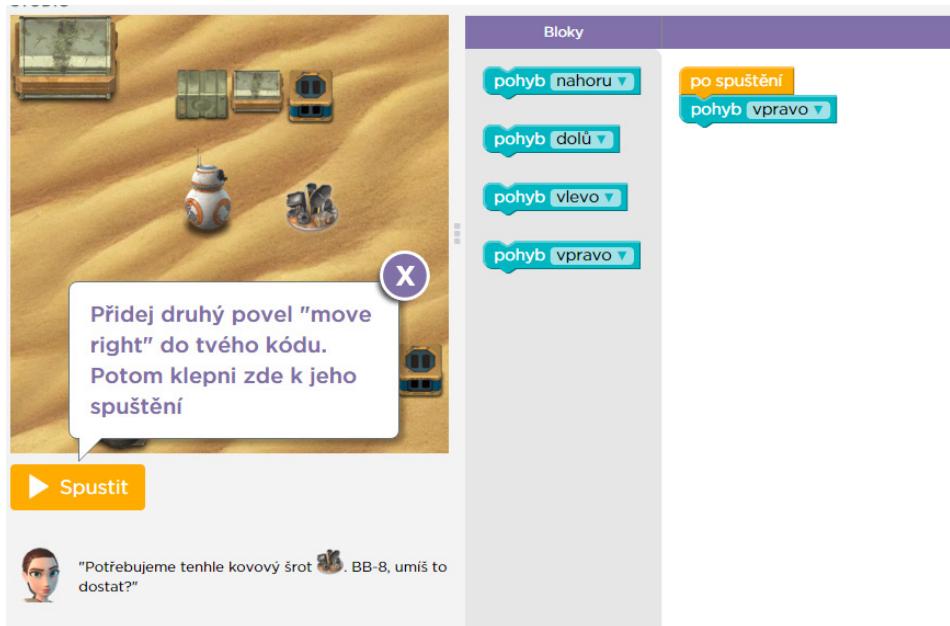
Platforma pomáhá dětem vyvíjet jejich algoritmické myšlení a programovací schopnosti zábavnou, intuitivní a originální cestou. Tynker Online poskytuje vlastní vizuální programovací jazyk, který se skládá z logických přetahovatelných bloků, které eliminují potřebu rozumět syntaxi programovacího jazyka. Vizuální jazyk je postaven na standardech otevřeného webu (HTML5, JavaScript, CSS) a funguje nezávisle na prohlížečích. Lze ho spustit i na mobilních platformách (iOS, Android). Jestliže děti dosáhnou určité úrovně, přejdou z vizuálního jazyka na JavaScript.

Lekce programování jsou koncipovány jako hry, které představují dětem úvod do programování. Zároveň je podporují v jejich kreativitě. Tynker je dostupný nejen na webu, ale i na iPadu a také na tabletích s operačním systémem Android.

4.5 Hour of Code

Hour of Code (Hodina kódu) je hodinový úvod do informatiky, který byl vytvořen k oslavě Týdne vyučování informatiky (7. – 13. prosince). Má za cíl přiblížit programování a informatiku zábavnou formou široké veřejnosti. Nepokládá si za cíl vychovat experty v informatice, ale snaží se ukázat, že informatika může být kreativní a zábavná. Hodinu kódu lze zorganizovat jak na školách, tak i v různých kroužcích. Kurz lze absolvovat samostatně. Existují zde kurzy určené pro určitou věkovou kategorii a různou úroveň zkušeností. Fungují také na všech zařízeních a prohlížečích.

Tuto akci organizuje veřejná nezisková organizace Code.org, která se věnuje zájmu o informatiku ve společnosti. Projekt podpořily i známé společnosti jako Microsoft, Apple či Amazon.



Obrázek 4.6: Ukázka jedné z lekcí Hodiny kódu na motivy nového dílu Hvězdných válek. [24]

Průvodci lekcemi se mohou stát oblíbené postavy ze světa Star Wars, počítačové hry Minecraft nebo Anna a Elsa z Frozen.

Pro většinu kurzů již existuje lokalizace do českého jazyka.

5 Použité technologie

5.1 Node.js

Node.js je open source, meziplatformní běhové prostředí pro serverové a síťové aplikace. Aplikace pro Node.js jsou napsány v programovacím jazyce JavaScript a mohou být spouštěny na operačních systémech OS X, Microsoft Windows i Linux. Využívá model architektury řízené událostmi (event-driven) a asynchronní vstupní/výstupní operace, která dopomáhá k optimalizaci propustnosti a škálovatelnosti aplikace. Tyto technologie se velice často používají pro aplikace, které pracují v reálném čase.

Pro provádění kódu je využit interpret Google V8 a drtivá většina modulů je napsaná právě v jazyku JavaScript. V malém množství je použit kód v C++, které poskytuje nutné zázemí (fronta událostí pro řízení příchozích událostí, obsluha vstupních a výstupních operací). Node.js obsahuje vestavěné knihovny, které dovolují aplikacím se chovat jako webové servery bez potřeby softwaru, jako např. Apache HTTP Server.

Tato technologie je využívaná známými servery jako například LinkedIn, PayPal, Yahoo! nebo firmou Microsoft.

Prvním představením Node.js byla verze pro operační systém Linux v roce 2009 na konferenci JSConf EU. Jeho tvůrce Ryan Dahl ze společnosti Joyent, která vývoj Node.js sponzorovala, byl inspirován progress barem pro upload na serveru Flickr, který slouží ke sdílení fotografií. Webový prohlížeč nedokázal zjistit, jak velkou část souboru se nahrála, a musel se dotazovat webového serveru. Dahl chtěl tento proces změnit a zjednodušit, a to se mu také podařilo. V lednu 2012 však odstoupil z postu hlavního manažera projektu zabývajícího se Node.js. [25]

V červenci 2011 se objevila první verze Node.js, která byla podporována operačním systémem Windows. Tato událost je výsledkem spolupráce firmy Joyent a Microsoft.

Ryan Dahl po několika neúspěšných pokusech s různými programovacími jazyky nakonec zvolil pro Node.js jazyk JavaScript, který neměl existující programové rozhraní (API) pro vstupní a výstupní operace (I/O). To mu dovolilo definovat pravidla pro asynchronní a událostmi řízené vstupní a výstupní operace. Node.js tak může transparentně udržovat mnoho spojení bez toho, aby musel odmítat nová příchozí spojení. [25]

Node.js je inspirován jinými modely, jako například Event Machine od jazyka Python a Twisted modelem programovacího jazyka Python. Rozdíl od ostatních modelů je implementace fronty událostí (event loop), která je vytvořena jako jazyk a ne jako knihovna. Tradiční modely používají blokovací volání, ale právě Node.js tato volání neobsahuje. Místo toho jednotlivé požadavky vstupují do fronty událostí po vykonání skriptu (tato technika je založená na běžném fungování JavaScriptu).

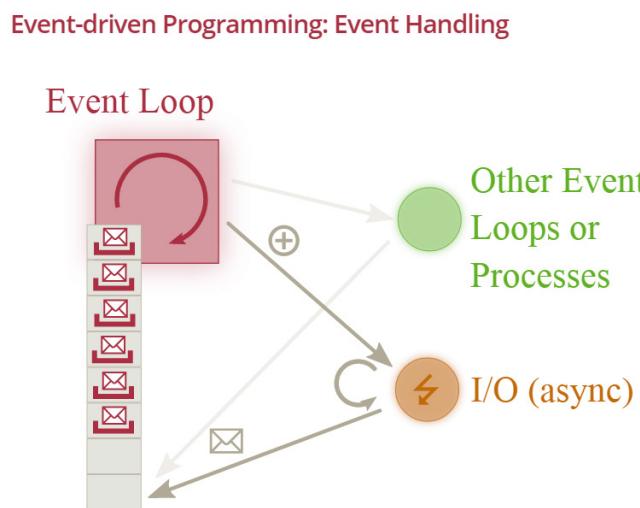
5.1.1 Práce s vlákny

Node.js aplikace jsou většinou jednovlákновé, ale přesto je možné používat ve verzi 0.10+ od JXcore více vláken.

Tradiční techniky webových serverů vyžadují při novém spojení vytvořit nové vlákno. Stále nová vlákna zabírají čím dál tím více RAM a může vést k jejímu vyčerpání. Node.js však pracuje jinak, protože pracuje na jediném vlákně, používá vstupní/výstupní volání bez blokování a dovoluje tak pracovat s desetitisíci souběžnými spojeními bez obav z vyčerpání RAM a ze složité režie přepínání mezi vlákny. Návrhový cíl ideální Node.js aplikace je, že každá funkce, která provede vstupní/výstupní operace, musí použít zpětné volání (callback). [26]

5.1.2 Architektura řízená událostmi (event-driven architecture)

Jedním za základů každé aplikace napsané v jazyce JavaScript je architektura řízená událostmi. Běžnou praxí je volání funkcí a posílání jejich návratových hodnot dalším funkcím. V JavaScriptu (potažmo v Node.js) existuje fronta událostí (události jako například příchozí HTTP požadavek, stisknutí tlačítka), které když nastanou, zavolá se na-



Obrázek 5.1: Schéma architektury řízené událostmi. [26]

stavená obslužná funkce (zpětné volání). Nemusí se tedy řešit synchronizace a události se řeší jedna po druhé.

Nevýhoda této architektury nastává v situaci, kdy některá funkce vyvolaná událostí trvá příliš dlouho a blokuje zpracování ostatních událostí. Klade se proto důraz na neblokující vstupní/výstupní operace a asynchronní zpracování. [26]

5.1.3 Správa balíků (npm)

Takzvaný npm je předinstalovaný správce balíků pro serverovou platformu Node.js. Je možné tak nainstalovat Node.js programy přímo z npm registrů. Díky dobré organizaci a jednoduché instalací programů třetích stran pomáhá npm vývojářům pracovat rychleji.

Balíky v npm registrech můžou být jak jednoduché pomocné knihovny, tak i složitější programy (například automatizovaný grunt.js, který řeší úkoly ve frontendu).

5.1.4 Porovnání Node.js s PHP

PHP je sice nejrozšířenější skriptovací programovací jazyk používaný pro programování dynamických webových stránek a webových aplikací, ale jeho prvenství je stále více ohrožováno právě Node.js.

Na začátku 21. století se jazyk PHP stal velmi oblíbený z několika důvodů. Mezi ně patří:

- jednoduchost – jednodušší než použití objektově orientovaného programování,
- skutečnost, že PHP je interpretovaný jazyk,
- možnost použít PHP přímo v HTML kódu pomocí značek <%php %>.

Postupem času se PHP stala obšírnou technologií, která je však zranitelná vůči bezpečnostním rizikům (např. SQL injection), neexistuje u ní žádný centralizovaný registr balíků, má nekonzistentní programové rozhraní (API) a podprůměrný výkon.

Díky svému stáří (první verze vznikla v roce 1995) má PHP výhodu v tom, že existuje již mnoho open-source aplikací (např. redakční systém WordPress), velké množství skriptů zdarma, kvalitních nástrojů a tištěných publikací.

5.1.4.1 Paradigma

Oba jazyky jsou funkcionální a s novější verzí PHP i objektově orientované.

5.1.4.2 Změny kontextu

Přecházení mezi různými prostředími a jazyky je často důvodem snížení efektivity při psaní softwarového kódu. [27]

Při použití PHP musí vývojář ovládat minimálně dva jazyky, kterými jsou PHP a SQL. Navíc musí samozřejmě ovládat HTML, CSS a JavaScript. Naopak u Node.js se dá snadno obejít bez PHP i SQL.

5.1.4.3 Moduly

U PHP existuje již už letitý repozitář PEAR, který instaluje balíky globálně na server a nebo také alternativa Composer. Vývojáři tedy musí hledat moduly na různých webových stránkách a spravovat je manuálně tím, že umísťují soubory s koncovkou .php do podadresářů jejich projektů.

Node.js má svého nadřazeného správce balíků npm, který je jednoduchý na ovládání a publikování. Vše je administrováno přes soubor package.json a lokálně aktualizováno.

5.1.4.4 Výkon

Jazyk PHP je sice relativně rychlý, ale kvůli problémovým místům v souborovém systému, databázi a kvůli požadavkům třetích stran se jeho výkon citelně snižuje.

Naproti tomu Node.js vítězí v rychlosti díky již zmíněnému Google Chrome V8 engine a neblokujícími I/O mechanismy.

5.2 Framework Express.js

Express.js (nebo také jen Express) Node.js je webový aplikační serverový framework pro tvoření jednostránkových, vícestránkových nebo hybridních webových aplikací. Podle autora Express.js TJ Holowaychuka je jeho framework inspirován webovým frameworkem a doménově specifickým jazykem Sinatra napsaným v Ruby, je tedy minimalistický, ale poskytuje mnoho funkcí ve formě pluginů. [28]

5.3 Embedded JavaScript

Embedded JavaScript (EJS) je jednoduchý šablonovací jazyk, který generuje značkování HTML s pomocí JavaScriptu. Poskytuje rychlou kompliaci a vykreslování a také jednoduché šablonovací značky (obrázek 5.2).

```
<% %>
```

Obrázek 5.2: Šablonovací značka EJS.

Zpřehledňuje kód, který rozděluje na data (ta jsou většinou zapsána v souboru .json) a šablony (s příponou .ejs). Je možné zobrazit i např. nějakou proměnnou, stačí zapsat značky na obrázku 5.3.

```
<%= content %>
```

Obrázek 5.3: Šablonovací značky EJS pro zobrazení dat.

Pokud se v datech objevuje značkování HTML, je pro správné zobrazování nutné napsat šablonovací značky ve tvaru na obrázku 5.4.

```
<%- content %>
```

Obrázek 5.4: Šablonovací značky EJS pro zobrazení dat se značkováním HTML.

5.4 JSON

JSON (JavaScript Object Notation – JavaScriptový objektový zápis) je otevřený standard formátu dat, který využívá člověkem dobře čitelný text pro přenos datových objektů založených na kolekcích párů název/hodnota nebo na seřazeném seznamu hod-

not. Vstupem může být jakákoli datová struktura (číslo, řetězec, pole, boolean, objekt), výstupem je vždy řetězec.

JSON je sice původně odvozený z jazyka JavaScript, ale pracuje jako na jazyku nezávislý datový formát. Kód pro syntaktickou analýzu nebo generování tohoto datového formátu je nativně přítomný v mnoha programovacích jazycích. [29]

Pro dobrou čitelnost je syntaxe tohoto datového formátu velmi jednoduchá. Objekt, který je v JSON chápán jako neuspořádaná množina párů název/hodnota, je uvozen hranatými závorkami. Každý název je od hodnoty oddělen dvojtečkou a každý pár je oddělen znakem čárky.

```
{  
    "index" : {  
        "header" : "<h2>Vítej u Ajťáčka</h2>",  
        "intro" : "<p>Vydej se s tučňákem Ajťáčkem na cestu</p>"  
    },  
  
    "piseme" : {  
        "header" : "<h2>Píšeme na klávesnici</h2>",  
        "intro" : "<p>Na klávesnici najdeme nejen písmena a...<p>",  
        "image_intro" : "/images/combinations.png",  
        "header_acc" : "<h3>Jak ale napišeme písmena s háčky?</h3>",  
        "accents" : "<p>Takové písmeno se musí napsat nadvavrát.<p>"  
    }  
}
```

Obrázek 5.5: Ukázka syntaxe kódu JSON.

5.5 jQuery

Javascriptová knihovna jQuery byla navržena pro zjednodušení skriptování HTML na straně klienta. Jedná se dnes o jednu z nejrozšířenějších knihoven, které se při vytváření webových stránek používají. jQuery je také svobodný a otevřený software pod licencí MIT.

Pomocí syntaxe jQuery se lépe orientuje v dokumentu, vybírají se DOM (objektově orientovaná reprezentace dokumentu) elementy, manipuluje s CSS vytváří se animace, řeší události a také je možné efektivněji vyvíjet Ajax aplikace. Vývojářům navíc dovoluje vytvářet javascriptové pluginy (např. abstrakce pro nízkoúrovňové interakce, pokročilé efekty atd.). Díky jednoduché selekci a manipulaci s DOM elementy (pomocí selektoru enginu Sizzle) vytvořilo jQuery nový styl programování, kde se setkávají algoritmy a datové struktury z DOM. A tak jQuery motivuje k rádnému rozdělení HTML (struktura) a Javascriptu (chování, interakce). [30]

Nejdůležitější částí kódu, kterou používá jQuery, je funkce `$()`, která je zkratkou pro jmenný prostor jQuery. Do funkce `$` se nejčastěji vkládá CSS selektor, který vybere dané elementy, a jQuery s nimi může dále pracovat pomocí různých metod. Funkce na obrázku 5.6 vybere element s id „score“ a pomocí metody `hide()` ho skryje.

```
$( '#score' ).hide();
```

Obrázek 5.6: Ukázka syntaxe kódu jQuery.

Do funkce \$ je možné kromě CSS selektorů vkládat i přímo HTML kód a přidávat tak nové DOM elementy za běhu, nebo také Javascriptové objekty.

5.6 HTML5

HTML5 je nejnovější verze značkovacího jazyka HTML používaného pro tvorbu webových stránek, která byla vydaná 28. října 2014 konsorciem W3C. Jeho hlavním úkolem je podpora nových multimédií při nahrazování statických webů weby dynamickými a sociálními. Klade také důraz na lehkou čitelnost člověkem, ale i na konzistentní zpracování různými zařízeními. Kromě nových funkcí HTML5 zahrnuje nejen HTML 4, ale také XHTML 1 a DOM Level 2 HTML.

Ve specifikaci skupiny WHATWG (Web Hypertext Application Working Group), která jako první začala HTML5 vyvíjet, staví nová verze jazyka HTML na principech kompatibility (zajištění spustilenosti i starého HTML kódu), přínosu (HTML5 jednoznačně upřednostňuje uživatele před programátory a nabízí mimo jiné i uvolněnější syntaxe, aby uživatel nepřišel o vykreslení stránky kvůli chybě v kódu. Snaží se také o čisté oddělení formy od obsahu, např. pomocí kaskádových stylů.), interoperability (obecné zjednodušení díky nativní podpoře ze strany prohlížečů, zjednodušenému DOCTYPE, zjednodušené znakové sadě a také díky mocnému a jednoduchému rozhraní API) a obecného přístupu (přístupnost uživatelům s postižením, nezávislost na médiu nebo podpora všech světových jazyků). [31]

HTML5 také zavádí mnoho nových elementů (včetně video, audio a canvas) a odstranění některých zastaralých, které suplovaly funkci kaskádových stylů. Mnoho nových elementů jsou tzv. sémantické značky, které zjednodušují návrh stránek HTML. Místo jednotlivého označování elementů DIV je možné v HTML5 přímo značky pro popisy sekcí (např. header, footer, section, nav). Na všechny tyto elementy lze samozřejmě aplikovat kaskádové styly. HTML5 kód pro vykreslení záhlaví webu může tedy například vypadat takto:

```
<section>
  <article>
    <header>
      <h3>Záhlaví článku</h3>
    </header>
    <p>
      ...
    </p>
  </article>
</section>
<footer>
  <h3>Zápatí článku</h3>
</footer>
```

Obrázek 5.7: Ukázka použití nových značek HTML5.

5.6.1 Canvas

Canvas je rozhraní HTML5, které umožňuje dynamicky generovat a vykreslovat grafiku, grafy, obrázky a animace. Princip plátna původně představila společnost Apple ve WebKitu pro Mac OS X. Při návrhu specifikace od WHATWG firma Apple dokonce poukazovala na svá vlastnická práva, ale nakonec souhlasila s volnou licencí. Před existencí rozhraní Canvas bylo možné realizovat kreslení pouze skrze zásuvné moduly, např. Flash od firmy Adobe. [31]

Je-li ve stránce použit element Canvas, vytvoří se obdélníková oblast, která má výchozí velikost (300 x 150 pixelů). Může však mít libovolnou velikost a další atributy, např. fixní pozici. Počátek souřadnic plátna je v souřadnicích x = 0 a y = 0 v levém horním rohu. S elementem Canvas lze jakkoliv pracovat s pomocí JavaScriptu. Aby se ale mohl Canvas použít, musí se nejdříve získat jako kontext, na který lze aplikovat různé akce. Na úpravy plátna rozhraní Canvas lze pohlížet jako na databázové transakce (začíná se, provede se, dokončí se).

Rozhraní Canvas, i když je mocným nástrojem, se nesmí nadužívat a mělo by být použité pouze tam, kde je to opravdu nutné. Nemělo by se používat např. jako nadpisový element kvůli snížení výkonu a dodržení konvence.

Rozhraní Canvas podporují všechny současné webové prohlížeče kromě starších verzí Internet Explorera (starší než verze 9.0). I pro tyto starší verze nebo i pro nestandardní prohlížeče vznikají projekty, které se snaží nabídnout podporu plátna.

5.7 Adobe Flash Professional CC a HTML5 Canvas

Program Adobe Flash Professional je využíván hlavně pro tvorbu vektorové grafiky a animací a jejich publikování pro webové stránky, webové aplikace, pro mobilní platformy nebo počítačové hry. Flash také podporuje rastrovou grafiku, formátovaný text, vkládání audia a videa a také skriptování pomocí vlastního jazyka ActionScript. Obsah může být publikován pro Flash Player, Adobe Air a také nově pro HTML5 a WebGL.

Flash poskytuje vlastní prostředí a nástroje pro vytváření grafiky, ale dovoluje také import zdrojových souborů z ostatních programů od společnosti Adobe a samozřejmě také ostatní standardní formáty obrázků, audia a videa. Pro animaci slouží časová osa s jednotlivými snímky a celá řada mocných nástrojů včetně doplnění pohybů nebo aplikace kosterního systému.

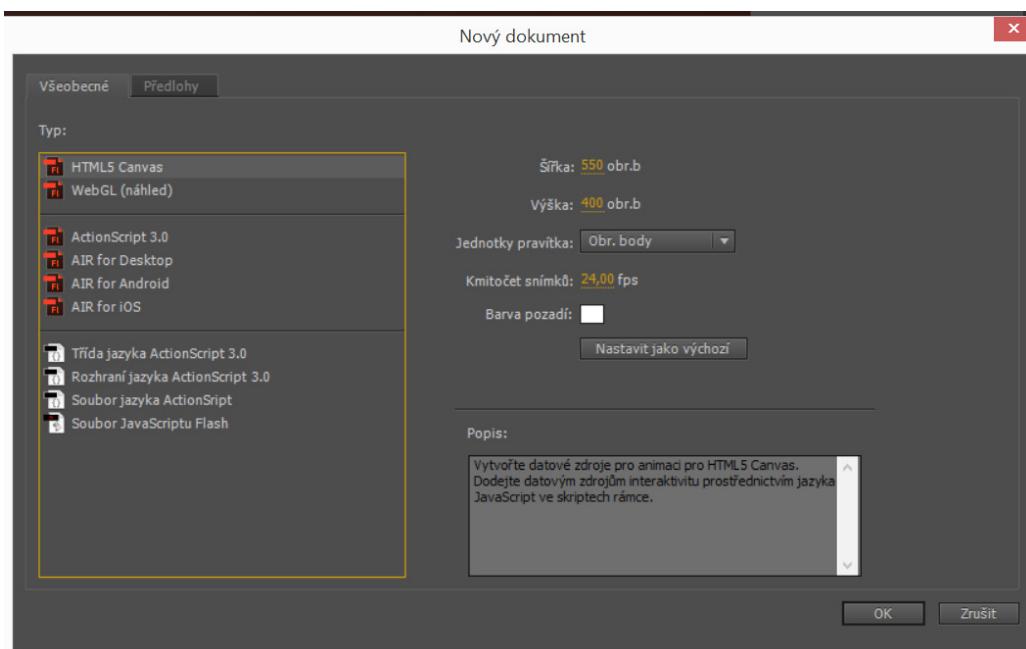
První verze této aplikace byla vydána v roce 1996 pod názvem FutureSplash Animator malou softwarovou firmou FutureWave Software. Tato verze také podporovala vektorovou grafiku a vektorové animace. V prosinci 1996 koupila firmu FutureWave firma Macromedia a program znova vydala pod názvem Macromedia Flash v1.0. V roce 2005 odkoupila společnost Macromedia firma Adobe Systems, která pokračovala ve vývoji tohoto softwaru až do nynější verze, tedy Adobe Flash Professional CC. [31]

V nejnovější verzi CC (Creative Cloud) lze vytvářet dokument jako HTML5 Canvas, který zajišťuje bohatý a interaktivní obsah HTML5. Pomocí časové osy, pracovní plochy

a dalších nástrojů z předchozích verzí aplikace Flash (některé nástroje nelze použít, protože je element Canvas nepodporuje, např. 3D transformace) lze vytvářet obsah, kde výstup bude generovaný právě v HTML5. Verze CC je integrována s rozhraním knihoven JavaScriptu CreateJS, díky kterému lze pracovat s bohatým interaktivním obsahem v otevřených webových technologiích prostřednictvím HTML5. Aplikace poté vygeneruje kód HTML a JavaScript pro obsah vytvořený na pracovní ploše pomocí rozhraní API Canvas, které provádí mapováním prvků v aplikaci Flash 1:1. Výstup je poté možné přehrát v jakémkoliv prohlížeči, který podporuje element Canvas. [32]

1. prosince 2015 Adobe oznámilo, že při další velké aktualizaci programu bude Flash nést nové jméno Adobe Animate. Tato změna je z důvodu upouštění od používání zásuvného modulu Adobe Flash Player na webu a podporuje webové standardy a zvyšující se oblibu a používání HTML5. [33]

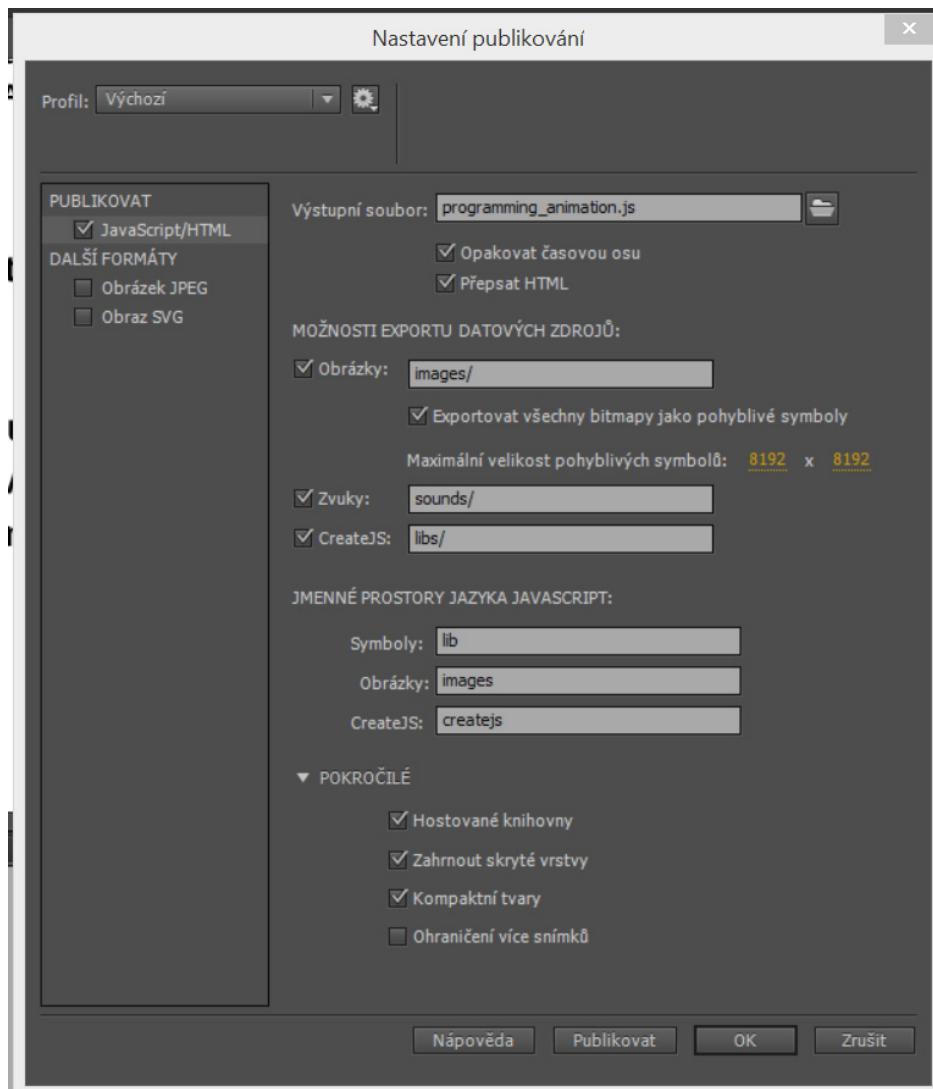
Vytvoření dokumentu aplikace Flash, který podporuje element Canvas, je velice jednoduché. Stačí zvolit v menu Soubor > Nový... a vybrat typ HTML5 Canvas a zvolit velikost plátna, barvu pozadí, jednotky pravítka (nejlépe nechat v obrazových bodech kvůli stejnemu měřítku, které se používá v kaskádových stylech) a kmitočet, který nastavuje počet snímků, které se přehrají za jednu sekundu.



Obrázek 5.8: Možnosti při vytváření nového dokumentu HTML5 Canvas.

Stejně jednoduché je i publikování obsahu v HTML5. V menu je vybrat položku Soubor > Nastavení publikování. Jako výchozí složka je označena složka se zdrojovým souborem .fla, lze ji ale samozřejmě změnit. Je možné také zakázat opakování časové osy (na konci časové osy se animace zastaví) nebo zakázat publikování souboru HTML (vygenerují se pouze datové zdroje a JavaScriptový soubor). Možnosti exportu datových zdrojů označují relativní adresy pro export obrázků, zvuků a pomocných knihoven

v rozhraní CreateJS. Je možné také nastavit jmenné prostory, ze kterých jsou pak odkazovány symboly, obrázky a knihovny CreateJS. Pokud je možnost „Hostované knihovny“ vybrána, použijí se kopie knihoven hostovaných v síti CDN (Content Delivery Network) na webu code.createjs.com. Knihovny se pak ukládají do vyrovnávací paměti. Pokud se ve skrytých vrstvách nenachází obsah nutný k běhu projektu, je dobré je nezahrnovat do publikování kvůli zmenšení velikosti výsledných souborů. Kompaktní tvary jsou výstupy vektorové grafiky uvedené v kompaktním formátu nečitelném pro člověka. Možnost ohraničení snímků znamená, že každý snímek časové osy bude obsahovat vlastnost frameBounds, která obsahuje libovolné pole obdélníků odpovídajících ohrazením jednotlivých snímků.



Obrázek 5.9: Možnosti pro publikování projektu HTML5 Canvas.

Publikovaný výstup obsahuje dva soubory – HTML a JavaScript. V souboru HTML se nachází definice elementu Canvas, vyvolává jmenný prostor rozhraní CreateJS a odpovídající JavaScriptový soubor, který obsahuje definice všech prvků a jejich vlastností a kód představující jejich interaktivitu, řešení událostí apod.

5.7.1 CreateJS

CreateJS je sada modulárních knihoven a nástrojů JavaScriptu, které v aplikaci Flash mohou manipulovat s obsahem prostřednictvím HTML5. Sadu knihoven tvoří EaselJS, TweenJS, SoundJS a PreloadJS. Psaní kódu je možné přímo v aplikaci Flash v panelu Akce uvnitř jednotlivých klíčových snímků. V kódu je také možné pracovat přímo s určitými vytvořenými objekty na pracovní ploše nebo jejich referencemi uvnitř aplikace. Pro zvýšení efektivity programování panel Akce, kde je možné psát kód jazyka JavaScript, obsahuje několik užitečných funkcí: doplňování kódu (při psaní znaků aplikace zobrazí seznam návrhů, které by mohly dokončit daný řetězec), zvýraznění syntaxe, barvení kódu, nebo automatické doplňování závorek. Je také možné využít fragmenty kódu, které představují základní interakce a manipulace s objekty, a ovládání událostí. [33]

Knihovna EaselJS je užitečná pro vytváření her, generativního umění a dalších grafických a animačních projektů. Poskytuje přímočará řešení pro práci s bohatou grafikou a interaktivitou s elementem Canvas. Poskytuje rozhraní, které je velmi podobné ActionScriptu aplikace Flash, ale zahrnuje citlivost jazyka JavaScript. Skládá se z plného hierarchického seznamu zobrazení, interakčního modelu a řady pomocných tříd pro snadnou práci s elementem Canvas. [35]

Jednoduchá ale mocná knihovna TweenJS slouží pro doplňování a animování v HTML5 a JavaScriptu. Funguje jak samotná, tak s integrací s EaselJS. Je vytvořena tak, aby mohla pracovat jak s numerickými vlastnostmi objektů, ale také s kaskádovými stylami. Rozhraní je velice jednoduché, ale je možné díky němu vytvářet složité animace pomocí řetězení příkazů. [35]

SoundJS je knihovna, která poskytuje jednoduché rozhraní pro práci se zvukem. Může pracovat s knihovnou PreloadJS pro nahrávání audio souborů. [35]

Samotná knihovna PreloadJS slouží pro přednahrání různých assetů jako jsou obrázky, zvuky, JavaScript, data aj. Používá XMLHttpRequest Level 2 (poskytuje funkcionality pro přenášení dat mezi klientem a serverem). [35]

5.8 Responzivní web design

Pojem responzivní web design poprvé vyslovil a popsal americký programátor Ethan Marcotte v článku na blogu A List Apart v roce 2010. Inspiroval se myšlenkou responzivní architektury, kdy fyzické prostory reagují na přítomnost procházejících lidí. Architekti tak mohou experimentovat s uměleckými instalacemi a se systémy zdí, které se mohou s pomocí pohybových senzorů ohýbat, otáčet, přesouvat, aby se vyhnuly případné kolizi s člověkem.

První přizpůsobení vzhledu webové stránky pro různá zařízení bylo již možné v CSS 2.1. Bylo možné přiřadit konkrétní kaskádový styl pro dané zařízení přes tzv. media types. Tento přístup byl a je používán k tisku stránek na webu, kdy se odstraní navigace a další prvky, které nemají na tištěném papíře žádný význam. [36]

```
<link rel="stylesheet" type="text/css" href="print.css" media="print"/>
```

Obrázek 5.10: Přiřazení kaskádového stylu pro konkrétní účel (tisk).

World Wide Web Consortium (W3C) však přišlo ve specifikaci CSS3 s tzv. media queries, která vylepšovala media types. Media query může sledovat nejen typ zařízení, ale může kontrolovat i fyzické charakteristiky zařízení, které vykresluje danou stránku. Media query se skládá ze dvou komponent – typu média a vlastního dotazu na vlastnost média, který je následován vlastní hodnotou. [36]

```
@media only screen and (min-width:660px) and (max-width:952px){  
    ...  
}
```

Obrázek 5.11: Media query vymezuje změny v kaskádovém stylu podle zařízení a jeho velikosti.

Jsou dva přístupy k responzivnímu návrhu. První (tzv. liquid layout) zahrnuje neutálé přizpůsobování se v jednotlivých krocích změně velikosti uživatelského prohlížeče. Druhý (tzv. adaptive layout) reaguje na změny velikosti okna prohlížeče pouze v určitých fixních bodech (např. změna počtu sloupců). Tyto dva přístupy se také často kombinují. [37]

Ovládání aplikací na počítači je však značně odlišné od dotykových zařízení, odlišný je i způsob interakce. Některé aplikace pro desktopová zařízení nemusí mít pro dotyková zařízení smysl a naopak. [38]

6 Tvorba interaktivní učebnice

6.1 Hosting pro Node.js

Na Node.js je založeno čím dál více aplikací různých firem včetně Netflix, Walmart nebo PayPal. Co se týče hostingu je zde však problém pro jednotlivce nebo malé firmy najít poskytovatele. V zahraničí sice existuje několik profesionálních platforem pro hostování Node.js aplikací, ale v České republice se provozovatelé webhostingu k této technologii staví většinou zády.

6.1.1 Virtuální privátní servery

Alternativou specializových hostingů pro Node.js jsou tzv. virtuální privátní servery (VPS). Zde můžete nahrát libovolnou linuxovou distribuci, jsou dostupná všechna práva a uživatel se o server stará sám (zajištění trvalého běhu aplikace, restart serveru při chybě nebo při změně skriptu). VPS se však hodí pouze hlavně na vývoj a testování. Např. při přecházení na novou verzi Node.js je možné si vyzkoušet upgrade na VPS, a když je vše v pořádku, nasadit ji specializovaný hosting. Pro reálný provoz není doporučený z důvodu nespolehlivosti.

6.1.2 OpenShift

Jedna z platforem, která je dostupná i zdarma, je OpenShift od společnosti Red-Hat. Má již několik milionů uživatelů po celém světě. Cenové programy se zde řídí podle toho, kolik služeb chce uživatel využívat (na začátku po registraci má několik služeb zdarma). OpenShift kromě Node.js aplikací umožňuje hostovat i aplikace napsané v jazycích Java a PHP. Uživatel má k dispozici i několik kontejnerů s databázemi jak s SQL, tak i No-SQL. Nasazování probíhá pomocí gitu.

6.1.3 Heroku

Heroku je profesionální cloud hosting, kde běží několik milionů aplikací. Každý uživatel používá vlastního klienta, pomocí kterého ovládá svůj účet, a soubory se zde nahrávají také pomocí gitu. Mimo hostingu pro Node.js Heroku poskytuje prostor i pro Ruby, Java, Python a další aplikace.

Tento hosting poskytuje velké množství doplňků databáze.

6.1.4 Roští.cz

V Česku zatím existuje jediný poskytovatel a tím je Roští.cz zaměřující se na hostování právě Node.js a také Python, PHP nebo Ruby aplikací. Placení funguje na bázi kreditů, které se denně spotřebovávají podle toho, jaké služby uživatel využívá a jak jeho aplikace vytěžují server. Každá aplikace je zde jako jeden kontejner (instance), který funguje podobně jako VPS. Tento kontejner využívá zdroje základního serveru, na kterém běží. Výhodou je dostatečná velikost RAM, která je nastavena pro každý jednotlivý kontejner od 64 MB až do 1024 MB RAM. Další výhodou je FTP a SFTP přístup, který u platforem hostující Node.js aplikace není běžný. Přístup přes SSH (např. pomocí SSH klienta PuTTY) je zde přítomen samozřejmě také z nutnosti manipulace s konzolí kontejneru.

Pro aplikaci učebnice byl vybrán právě tento hosting, kde uživatelská podpora probíhá téměř nepřetržitě. Aplikace musí naslouchat na portu 8000 a komunikovat HTTP protokolem. Aby změna kódu v Node.js nabrala platnosti, musí se vždy aplikace restartovat – buď pomocí tlačítka v administraci aplikace na webu hostingu, nebo pomocí příkazu na obrázku 6.1 zadaného do konzole.

```
supervisorctl restart app
```

Obrázek 6.1: Příkaz pro restart aplikace na hostingu Roští.cz.

6.2 Express generator

Framework Express.js pro Node.js nabízí možnost použití svého generátoru, který rychle vytvoří základní kostru aplikace. Instalace generátoru se provádí jednoduchým příkazem v konzoli po připojení na server pomocí SSH.

```
$ npm install -g express-generator
```

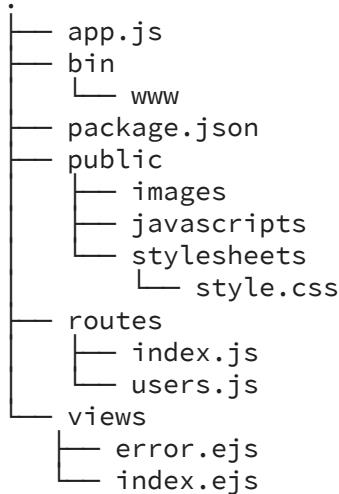
Obrázek 6.2: Příkaz pro nainstalování balíku s Express generátorem.

Poté je možné pomocí příkazu `express -h` zobrazit možnosti dalších příkazů (např. zjistit verzi Express.js nebo nastavit šablonovací systém). Po zadání příkazu na obrázku 6.3 se vytvoří základ webové Express aplikace s názvem "ajtacek" v aktivním adresáři používající šablonovací systém EJS.

```
$ express -e ajtacek
```

Obrázek 6.3: Příkaz pro vygenerování kostry aplikace s názvem ajtacek.

Express generator vytvoří systém souborů zorganizovaných do jednotlivých složek, která představuje jednoduchou Express aplikaci.



Obrázek 6.4: Vygenerovaná adresářová struktura.

Ve složce bin jsou umístovány všechny skripty potřebné pro start aplikace (v tomto případě soubor www, kde pro Roští.cz se musí změnit číslo naslouchajícího portu), včetně definice portu. Veškerý middleware je poté spravován v souboru app.js a ve složce routes, kde se nachází všechno směrování. Složka public poté slouží pro ukládání souborů potřebných pro frontend (skripty, kaskádové styly, obrázky apod.). Adresář views obsahuje všechny soubory zvoleného šablonovacího systému.

Po vygenerování kostry aplikace je nutné také nainstalovat všechny nové závislosti (aktualizovat soubor package.json), protože Express generator nainstaloval nové moduly. Závislosti se mohou jednoduše nainstalovat v adresáři aplikace pomocí příkazu na obrázku 6.5.

```
$ npm install
```

Obrázek 6.5: Příkaz pro nainstalování závislostí.

Generátor přidal také některé moduly z npm například Morgan (modul pro zajištění logování) nebo Serve-favicon, díky kterému lze jednoduše přidat ikonu stránky (favicon.ico) do určitého adresáře bez nutnosti přidávání kódu do hlavičky webu. Stačí přidat do app.js řádek kódu na obrázku 6.6, který definuje místo, kde se ikona nachází.

```
app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
```

Obrázek 6.6: Kód, který definuje cestu k souboru favicon.ico.

6.3 Směrování (routing)

Směrování (routing) definuje koncové body aplikace (URI nebo cesta a určitá metoda požadavku HTTP protokolu, např. get, post atd.) a způsob, jakým budou reagovat na

klientské požadavky. Každý route obsahuje jednu nebo více funkcí, které se provedou při spojení nebo v případě, když nastane daná událost. Tato technika je nejdůležitější funkctionalita pro webovou aplikaci učebnice. Obecná definice směrování v Express.js může vypadat například jako na obrázku 6.7.

```
app.METHOD(PATH, HANDLER)
```

Obrázek 6.7: Obecná definice směrování.

App je instance Express.js, METHOD je metoda požadavku protokolu HTTP, PATH je cesta na server a HANDLER jsou funkce, které se mají provést při odpovídajícím routeu.

V aplikaci učebnice je využívána hlavně metoda get, díky které webový prohlížeč (klient) může získat data ze serveru.

```
router.get('/programujeme/programovacihra', function(req, res) {  
    res.render('programovacihra', {  
        title: 'Programovací hra',  
        chapter_name: 'programujeme'  
    });  
});
```

Obrázek 6.8: Konkrétní příklad směrování.

Jako odpověď serveru (objekt res) na požadavek (objekt req) je na obrázku 6.8 zadefinovaná metoda render, která vykreslí danou šablonu ze složky views a je možné v ní zadefinovat i lokální proměnné (v tomto případě název stránky a kapitoly). Častá chyba v syntaxi může být v poslední definované proměnné, za kterou nesmí být čárka.

Instance Express.js je na obrázku 6.8 nahrazena instancí třídy express.Router, která jako modul vytváří kompletní middleware a směrovací systém. Tento modul s routy musí být definován i v samotné aplikaci v souoboru app.js.

```
var routes = require('./routes/index');  
app.use('/', routes);
```

Obrázek 6.9: Určení cesty k routům a jejich definice.

6.4 Struktura aplikace

Učebnice je rozdělena do pěti kapitol, kde každá kapitola v sobě obsahuje odkaz na vědomostní kvíz nebo hru. Kromě odkazu obsahují textové vysvětlení různých pojmu v informatice doplněné o ilustrační obrázky nebo informační grafikou. Uprostřed každé kapitoly se nachází interaktivní grafika, se kterou děti mohou pracovat podle svého vlastního tempa.

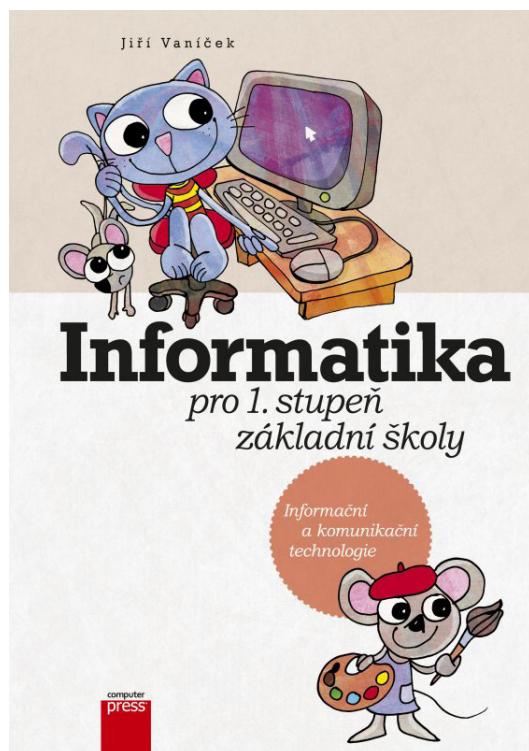
Většina animací je tvořena metodou frame-by-frame (animace po jednotlivých snímcích) při kmitočtu 12 snímků za sekundu. Tato stylizovaná ruční animace dobře koresponduje s nastaveným stylizovaným vizuálem aplikace.

Kromě jednotlivých kapitol a her se aplikace také skládá z úvodní stránky, kde je dětem představena učebnice a její maskot tučňák Ajtáček a myšky.

Pro informaci o učebnici a o autorovi je zde také zařazena podstránka „O aplikaci“, kde odkaz není tak dobře viditelný, aby nerušil děti při práci s aplikací.

6.5 Učebnice pro základní školy

Většina textového obsahu a i některé náměty na hry jsou převzaty z učebnice „Informatika pro 1. stupeň základní školy“ od Jiřího Vaníčka, která byla dne 6. 12. 2012 schválena Ministerstvem školství, mládeže a tělovýchovy a zařazena s šestiletou dobou platnosti do seznamu učebnic pro základní školy pro výuku oboru Informační a komunikační technologie. Je určena především pro žáky 4. a 5. ročníku 1. stupně základních škol, ale úvodní kapitoly zvládnou i menší děti. [39]



Obrázek 6.10: Obálka učebnice. [39]

Učebnice je bohatě ilustrovaná a obsahuje desítky úkolů, hříček a hádanek z různých odvětví informačních technologií (vytváření digitálních obrázků, ovládání počítače a jeho periferií, vyhledávání a zpracování informací atd.) a pro výuku základů programování využívá aplikaci EasyLogo.

Celou učebnicí provází postavička kocoura, který učí další postavičku myšku, jak zacházet s počítačem. V knize se objevují i další postavičky, jako například včela, vlk nebo medvěd. Publikace zachovává svůj vizuální styl ve většině svého rozsahu, nachází se zde ale například nekvalitní fotografie, které by mohly být také prezentovány

ilustrací nebo se zde bohužejí objevují gramatické nebo tiskové chyby, které brání pochopení dané látky nebo vyřešení příkladu.



Obrázek 6.11: Text v bublinách je totožný. [39]

NE	DÍKY	DÍKY
NIKDO	LE	SVÉ
SMRTELNÝM	SE	NOSTI

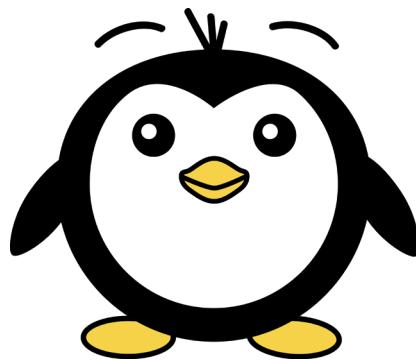
Obrázek 6.12: Kvůli stejnemu textu ve dvou různých buňkách tabulky nelze šifru vyřešit. [39]

6.6 Grafický návrh aplikace

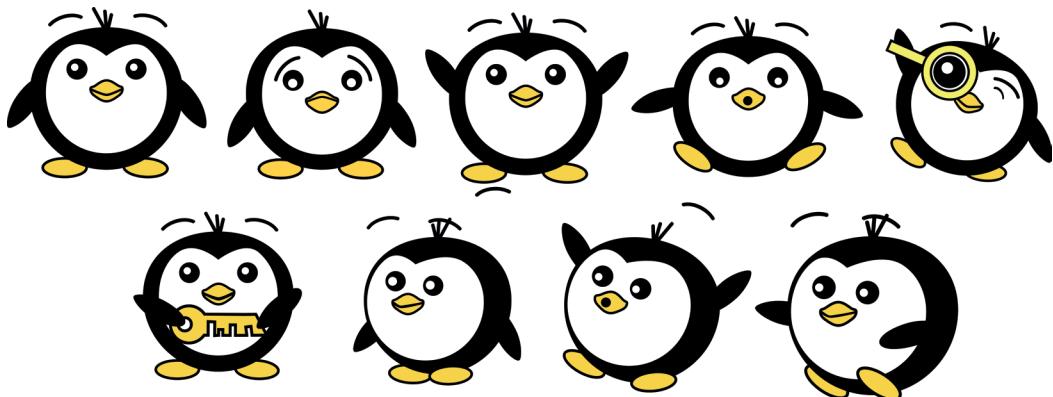
Nejdůležitějším motivem aplikace je hlavní maskot učebnice tučňáček Ajťáček. Postavička tučňáka byla vybrána proto, aby průvodce učebnicí, kterého by měla postava představovat, byla živá bytost a aby měla nějakou spojitost s informatikou. Samotný Ajťáček totiž představuje reminiscenci na tučňáka Tuxe, který je maskotem operačního systému Linux. Pro děti navíc může být atraktivní jeho roztomilost a jméno, ve kterém se vyskytuje rým a odkaz na IT (tučňáček Ajťáček).

Jako každý správný maskot je Ajťáček velice variabilní a může vyjadřovat různé emoce a interagovat s různými objekty. Postavička se tak může otáčet, sedět, být překvapená, smutná, zvědavá a zvládne i další pozice a emoce.

Ze samotného Ajťáčka vychází i barevná paleta, která je použita v celé učebnici, tedy černá, odstíny žluté a doplňková barva – modrá.

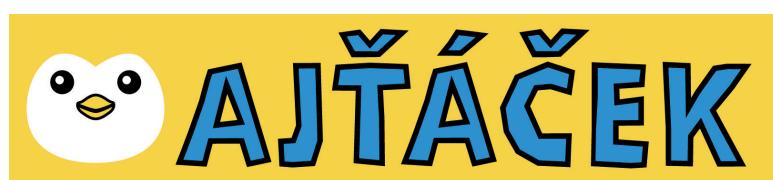


Obrázek 6.13: Hlavní maskot interaktivní učebnice tučňáček Ajtáček.



Obrázek 6.14: Ukázka různých pozic a emocí maskota Ajtáčka.

Jako jedna z variant značky slouží obličej Ajtáčka a název aplikace. Písmová varianta loga je odvozená z písma Open Sans, které je použité jako titulkové i textové písmo v různých řezech. Písmena jsou oproštěna od oblých tvarů a linie jsou pokřivené, aby vyjádřily hravost. Tento "hranatý" styl je poté použit i pro ostatní ilustrace, které představují neživé věci, například počítač a jeho komponenty, aby byl znatelný rozdíl mezi uživateli a technologiemi. Jedinou výjimkou je zeměkoule, která by po převedení na čtverec ztratila svůj vizuální význam.



Obrázek 6.15: Značka elektronické učebnice Ajtáček.

Aby mohl maskot Ajtáček interagovat nejen s počítačem, ale také s dalšími živými bytostmi, jsou vytvořeny další postavičky myšek, které stejně jako Ajtáček mají oblé tvary, aby vyjádřily svoji životnost. Myš jako postava byla zvolena jako slovní hříčka stejnojmenné počítačové periferie – počítačové myši.

6.7 Webdesign aplikace

Webový design učebnice byl zvolen jednoduchý a vzdušný pro lepší přehlednost a z důvodů vyniknutí ilustrací a interaktivních prvků. Během testování se grafický návrh aplikace mírně změnil ve formě menu. Důvody pro tento krok jsou vysvětleny v další části této práce.

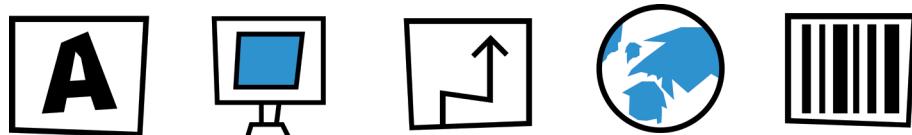


Obrázek 6.16: Prvotní grafický návrh učebnice.



Obrázek 6.17: Finální grafický návrh aplikace.

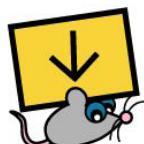
Každá položka menu, které má fixní pozici, má svůj vlastní piktogram, který se nese v duchu stylu písmové varianta loga, tedy žádné nerovné tahy nebo oblé tvary. Pro kapitolu o psaní na klávesnici je zvolen piktogram představující písmeno klávesnice, pro část o bezpečnosti práce s počítačem byl vybrán monitor. Pro kapitolu o základech programování byla zvolena klikatá šipka, která představuje stylizaci algoritmu popisující nějakou cestu. Lekce o internetu dostala piktogram se zeměkoulí představující „celosvětovost“ internetu. A konečně poslední kapitola o šifrách a kódech je prezentována stylizovaným čárovým kódem.



Obrázek 6.18: Piktogramy jednotlivých kapitol.

Hlavní obsah jednotlivých stránek je organizován do jednoho nebo dvou sloupců. Pokud je obrazovka zařízení příliš malá, dva sloupce se spojí v jeden. Pro děti je doporučeno používat větší a přehledné jednoduché písmo bez prvků, které by odváděly pozornost od kontextu dokumentu, proto je zvolený jednoduchý bezserifový font s názvem Open Sans. [40]

Na konci každé kapitoly se nachází opakující se prvek: ilustrace upozorňující na tlačítko a popis kvízu nebo hry a samotné tlačítko a popis.



Vyzkoušej své vědomosti v kvízu o bezpečnosti práce s počítačem!

KVÍZ O BEZPEČNOSTI

O aplikaci

Copyright © Jakub Špiřík

Obrázek 6.19: Postavička myšky upozorňuje na tlačítko s odkazem na hru.

6.8 Jednotlivé části učebnice

6.8.1 Úvodní stránka

Úvodní stránka aplikace Ajtáček představuje dětem v krátkém textu obsah a konцепci učebnice spolu s uvedením a seznámením se s maskoty aplikace.

Ve spodní části je použito rozhraní Canvas, kde se nachází stylizovaný statický obrázek monitoru se shlukem slov, která se vztahují k jednotlivým kapitolám nebo jejich částem. Samotný statický obrázek postačuje z hlediska estetické i informační hodnoty, a tak není ho nutné animovat nebo mu zbytečně přidávat interaktivitu. Jinak je tomu u postaviček, které samy od sebe mrkají a mávají a dávají tak signál, že jsou interaktivní. Po najetí myší na jednotlivou postavu se u daného maskota objeví textová bublina s pozdravem (myšky) nebo instrukcí (Ajtáček), kam má uživatel pokračovat.

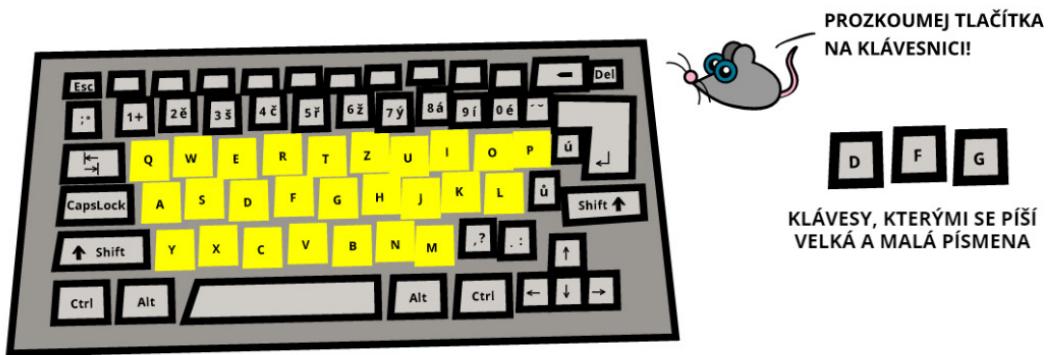
Klikni nahoře v menu na jednu z kapitol!



Obrázek 6.20: Interaktivní grafika na úvodní stránce učebnice.

6.8.2 Píšeme na klávesnici

Tato část se zabývá ukázkou rozložení tlačítek na klávesnici a vysvětlením jejich funkcí. Text doprovází ilustrace Ajtáčka a myšek, kteří drží klávesnice. Dále je zde informační grafika o psaní písmen s diakritikou a také část s interaktivní klávesnicí. Tento interaktivní prvek uvádí myška s vysvětlující textovou bublinou. Aby myška upoutala pozornost, vrtí ocáskem a mrká. Blikající klávesy indikují interaktivnost klávesnice. Jednotlivé skupiny kláves jsou aktivní a při najetí myší se zobrazí reprezentanti dané klávesy s vysvětlujícím textem.



Obrázek 6.21: Interaktivní informační grafika klávesnice.

6.8.3 Psací hra

V Psací hře je úkolem uživatele získat co nejvíce bodů za správné stisknutí daných písmen, která se ukáží na obrazovce. Hra je realizována s pomocí knihovny jQuery, kde je nejdůležitějším konceptem hlavní element, v kterém se vyměňují dva další elementy, přičemž je vždy viditelný pouze jeden z nich, aby mohlo být docíleno animace pohybu při výměně náhodně generovaného písmena.

Před začátkem samotné hry se v hlavním elementu nachází tlačítko Start a animace myšky vysvětlující pravidla. Po stisknutí tlačítka Start tlačítko i myška zmizí a objeví se odpočítávání, počítadlo skóre a první náhodně vygenerované písmeno (pro písmena je zvoleno serifové písmo, aby byly patrnější rozdíly například mezi velkým I a malým l). Náhodné písmeno se vybírá ze seznamu možných znaků (v tomto případě všechna malá a velká písmena bez diakritiky) pomocí metody Math.random(). Toto písmeno se uloží také do nové proměnné ve formě kódu Unicode pro účely porovnání vstupu z klávesnice.

```
function displayChar() {
    var rndmChar = "";
    var possible = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
    rndmChar = possible.charAt(Math.floor(Math.random()
        * possible.length));
    rndmCharCode = rndmChar.charCodeAt(0);
    $(stage).append('<div class="char">' + rndmChar + '</div>');
    ...
}
```

Obrázek 6.22: Generování náhodného písmena.

Pokud se stisknuté tlačítko neshoduje s vygenerovaným písmenem, jako zpětná vazba se písmeno zatřese pomocí metody `effect()`. Při správné odpovědi se přehodí elementy, kde na novém elementu je již vygenerované písmeno a ve starém elementu se smaže obsah. Při každé animaci se uzavírá zámek (proměnná `lock`), díky kterému se po době animace ignoruje vstup z klávesnice.

Po skončení odpočtu se objeví finální skóre s komentářem, s tlačítkem možného restartování hry a animací myšky. Pokud uživatel získal 25 a nebo více bodů, komentář ho povzbudí k prozkoumání dalších kapitol, jinak mu doporučí restartování hry.



Obrázek 6.23: Úvodní obrazovka Psací hry.

6.8.4 Bezpečně s počítačem

Kapitola Bezpečně s počítačem se zabývá nejen hygienou práce s počítačem, ale také bezpečností co se týče chování na internetu. Upozorňuje také na nebezpečnost virů.

Hlavní částí této stránky je animace pomocí rozhraní Canvas s dialogem postav Ajtáčka a myšky právě o bezpečnosti u počítače. Při jejich rozhovoru u počítače se objevuje postava, na které je ukázáno, jak je správné sedět. Dále reflekтуje dané problémy zmíněné v dialogu (viry, zlí lidé na internetu apod.). Přes dialog se uživatel naviguje pomocí žluté šipky, protože u dětí je dobré nechat uživatele nastavit své vlastní tempo a frekvenci interakce s aplikací. [16]

Po informační interaktivní grafice následuje krátká část s ilustračním obrázkem o vhodném umístění počítače v místnosti.



Obrázek 6.24: Animace o bezpečnosti práce s počítačem.

6.8.5 Kvíz o bezpečnosti s počítačem

Kvíz o bezpečnosti s počítačem sdílí vizuální i funkční podobnost s Psací hrou. Existuje zde také hlavní element, ve kterém jsou vnořené další dva, které vyměňují otázky. Data pro kvíz se získavají z JSON souboru, kde jsou uložené jednotlivé otázky, příslušné odpovědi (první odpověď je vždy správná, pro uživatele se potom jejich pořadí generuje náhodně) a obrázek k otázce. Jednotlivá data jsou získána pomocí jQuery metody `getJSON()`.

```
$.getJSON('/quiz/quiz_safety.json', function(data) {
    for(i=0;i<data.quizlist.length;i++){
        questionBank[i]=new Array();
        questionBank[i][0]=data.quizlist[i].question;
        questionBank[i][1]=data.quizlist[i].option1;
        questionBank[i][2]=data.quizlist[i].option2;
        questionBank[i][3]=data.quizlist[i].option3;
        questionBank[i][4]=data.quizlist[i].image;
    }
    numberofQuestions=questionBank.length;

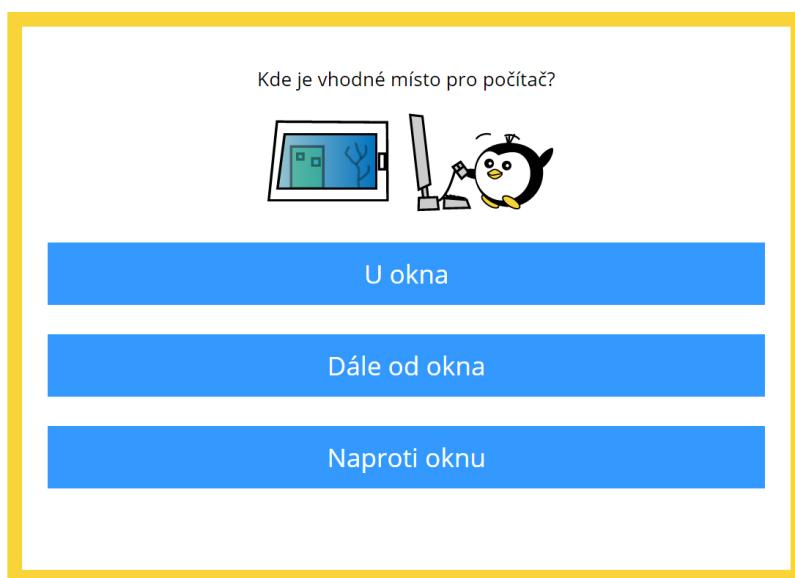
    displayQuestion();
})
```

Obrázek 6.25: Získání dat ze souboru JSON.

```
{"quizlist": [
    {
        "question": "Kde je vhodné místo pro počítač?",
        "option1": "Dále od okna",
        "option2": "Naproti oknu",
        "option3": "U okna",
        "image": "quiz_window.png"
    },
    ...
]}
```

Obrázek 6.26: Ukázka formátování dat otázek v JSON souboru.

Po zvolení odpovědi následuje ihned zpětná vazba v podobě červeného nápis „špatně“ nebo zeleného nápisu „správně“. Po zadání správné odpovědi se zobrazí výsledky, komentář k nim a také GIF animace Ajťáčka. Ten je podle výsledku buď smutný a krčí rameny, nebo je veselý a skáče. Při všech správně zadání správných otázkách vybízí komentář děti k pokračování v některé jiné kapitole.



Obrázek 6.27: Ukázka zobrazení otázky v kvízu o bezpečnosti práce s počítačem.

6.8.6 Programujeme

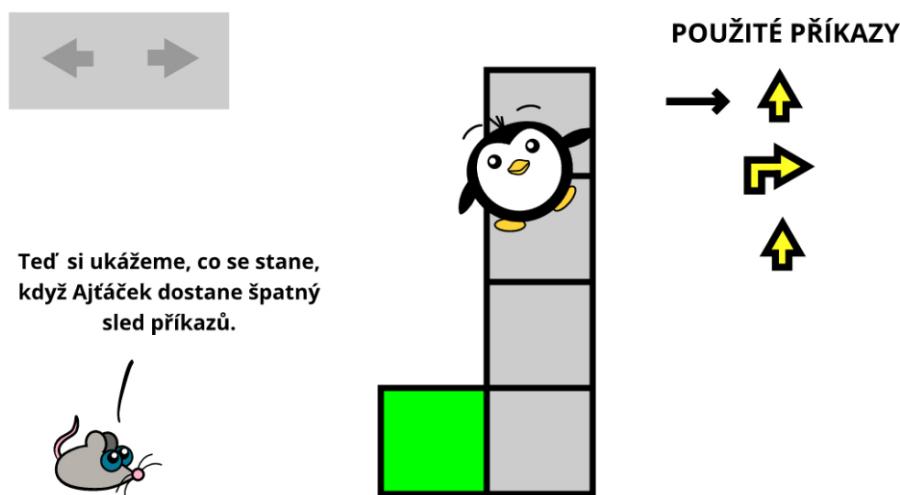
V kapitole o programování se děti dozvídají o základních stavebních kamenech programování a budou naváděny k algoritmickému myšlení.

V ukázce realizované pomocí rozhraní Canvas jsou představeny tři jednoduché programy (správný program, nesprávný program, ukázka opakování/jednoduchého cyklu) ve formě tří typů šipek (dopředu, doprava, doleva), které jsou komentovány myškou. Princip programu ve formě pohybu postavičky je převzat z mnoha dětských progra-

movacích jazyků jako například Karel, Logo nebo i programovacích nástrojů z Hour of Code. Vybírání a spouštění animací je opět realizováno pomocí šipek a děti si pouští programy podle vlastního tempa.

Tvorba krokových cyklů maskota Ajťáčka bude vysvětlena v následující části práce.

Na konci kapitoly je dětem uveden příklad zápisu programu „ze života“, konkrétně přípravy čaje, ve formě jednotlivých vět. Zápis doplňuje ilustrace Ajťáčka s rychlovarnou konvicí a hrnkem s čajem.



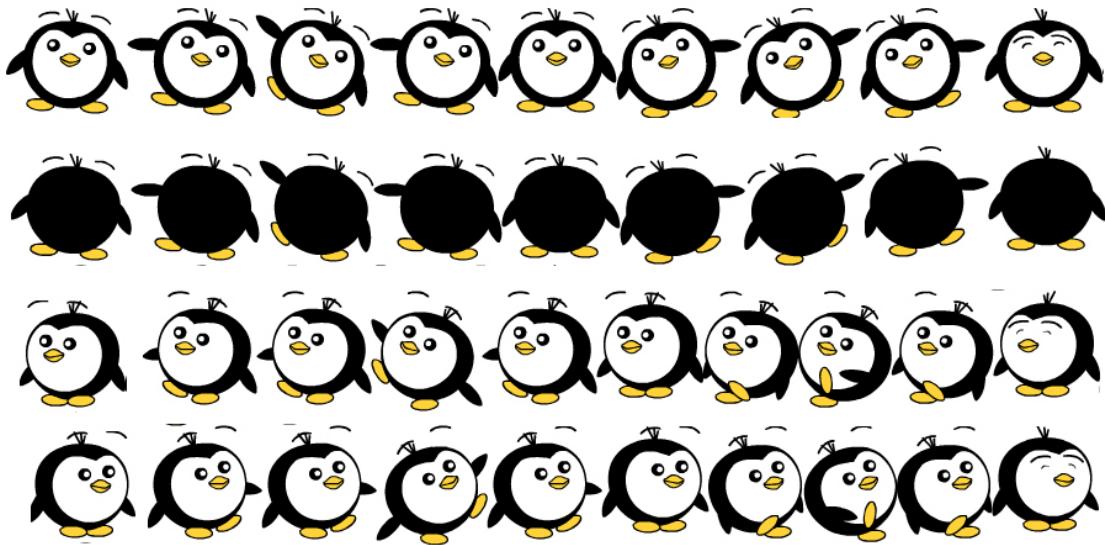
Obrázek 6.28: Ajťáček při provádění ukázkového programu.

6.8.7 Programovací hra

Programovací hra je realizována pomocí rozhraní Canvas. Skládá se z hracího pole ze čtvercových dlaždic, po kterých se může pohybovat postava Ajťáčka (mimo hrací pole se nelze posunout). Cílem hry je posbírat postupně písmena z hracího pole, aby utvořila slovo „sova“. Ajťáček se ovládá pomocí stejných příkazů (šipky vpřed, doleva, doprava) jako v programech v kapitole Programujeme a počet příkazů, které lze provést, je omezen na 35. Mimo hrací pole se zde nacházejí dvě myšky, které dodatečně vysvětlují ovládání, pokud si uživatel nepřečetl v úvodu hry.

Je-li sebráno nesprávné písmeno nebo pokud je vyčerpán počet povolených příkazů, objeví se dialog s tlačítkem restartu a animací smutného Ajťáčka. Pokud hráč sebere písmena ve správném pořadí, objeví se dialog s povzbuzením k prozkoumání dalších kapitol a veselého Ajťáčka s myškami.

Pro Ajťáčka je nutné vytvořit čtyři krokové cykly pro vertikální a horizontální, z toho pohyby směrem vlevo a vpravo se mohou ozrcadlit.



Obrázek 6.29: Sada krokových cyklů maskota Ajtáčka.

Směr, kterým je Ajtáček natočený, řídí proměnná direction, tato proměnná se tedy mění při stisknutí šipek doleva a doprava. Při pohybu dopředu je zvolen ve funkci moveForward() daný směr. Poté se zjistí, jestli se Ajtáček nebude nacházet mimo hrací plochu, spočítá se nová pozice a ve filmovém klipu player, který je v kódu reprezentován referencí pl, se přejde na snímek s označením left_anim, kde se nachází krokový cyklus.

```
if (direction == "west") {
    if (pl.x - tileSize > tileX) {
        var position = {
            x: pl.x - tileSize,
            y: pl.y
        };
        pl.gotoAndPlay("left_anim");
    }
}
```

Obrázek 6.30: Definice směru, kterým se bude Ajtáček pohybovat.

Samotné doplnění pohybu je realizováno pomocí třídy Tween knihovny TweenJS, kde se řetězí metody get(), která získá instanci hráče, a to(), která definuje cílovou pozici a čas, za který se objekt dostane do této pozice.

```
var tween = createjs.Tween.get(pl).to(position, 500);
```

Obrázek 6.31: Kód doplnění pohybu.

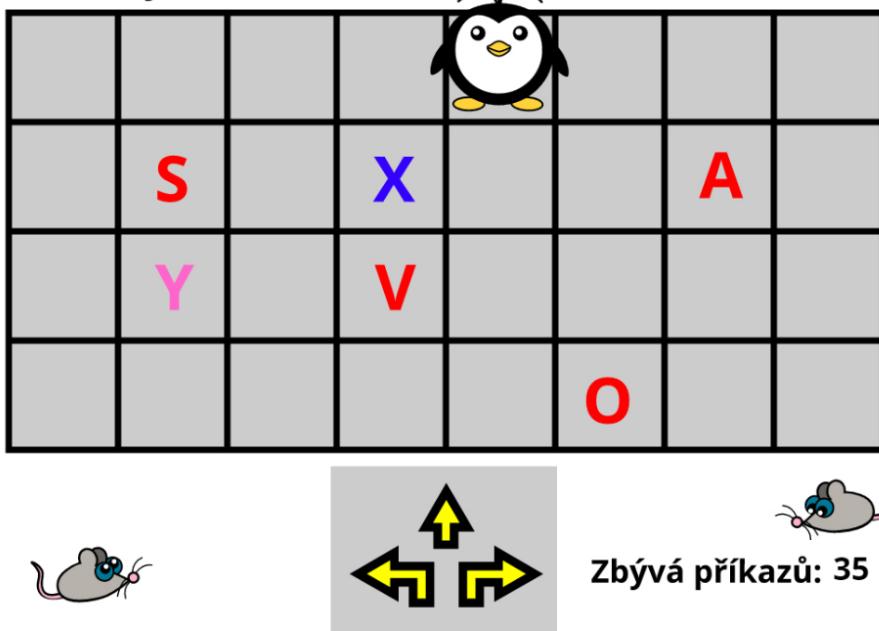
Stejně jako u psací hry nebo kvízu zde existuje zámek, který po dobu animace znemožňuje přijímat událost kliknutí na šipky.

Detekce kolizí u písmen je realizována pomocí třídy Ticker knihovny EaselJS, která představuje nastavení tikotu (časový interval), kdy naslouchající může naslouchat události tick, která ho upozorní vždy po uplynutí daného časového intervalu. Funkce collision() zjistí, jestli objekty pl (reference hráče) a charV (reference písmena) se překrývají (čísla za objekty značí velikosti daných objektů). Odstraní se naslouchání a zjistí se, jestli je písmeno sebráno ve správném pořadí. Pokud ano, písmeno se skryje (v případě posledního písmene se přejde na snímek s dialogem výhry), pokud ne, přejde se na snímek s dialogem o nesprávně sebraném písmenu.

```
charV.addEventListener("tick", hitTestCharV.bind(this));
function hitTestCharV() {
    if (collision(pl, charV, 50, 50, 80, 80)) {
        charS.removeEventListener("tick", hitTestCharV.bind(thisFrame));
        if(!(charS.visible) && !(charO.visible)) {
            charV.visible = false;
        }else{
            thisFrame.gotoAndPlay("lostBadChar");
        }
    }
}
```

Obrázek 6.32: Řešení události detekce kolize.

Pomoz Ajťáčkovi sestavit slovo SOVA!



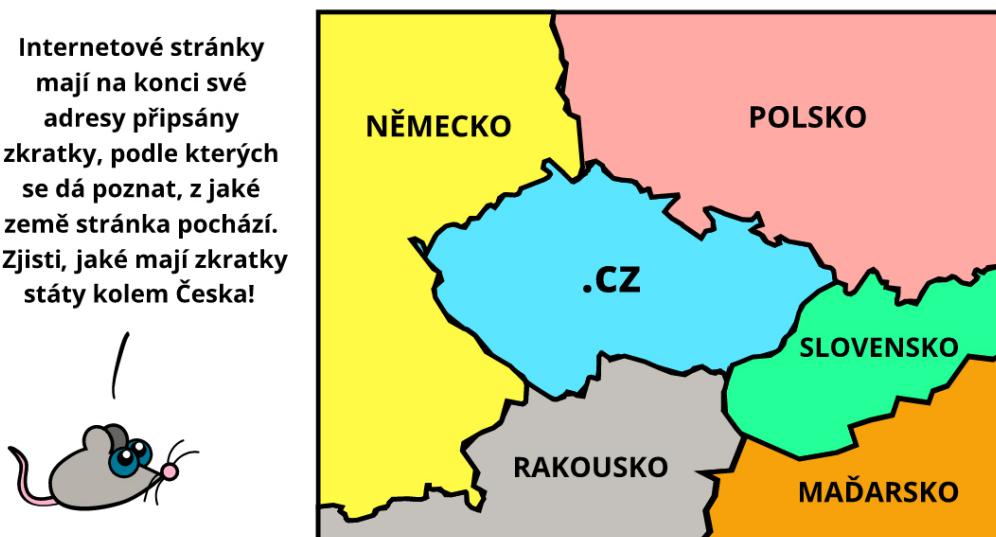
Obrázek 6.33: Ukázka herního pole.

6.8.8 O internetu

V kapitole „O internetu“ je dětem zjednodušeně představena síť internet. Ilustrační obrázek představuje myšky s počítači jako uživatele právě připojené k internetu. Následuje část o internetové adrese a informační grafika s myškami, která vysvětuje jednotlivé části internetové adresy.

Uprostřed kapitoly se opět nachází element Canvas s interaktivní grafikou mapy blízkých států České republiky, která odhaluje jednotlivé zkratky daných států v jejich internetových adresách. Na interaktivitu opět upozorňuje myška, která v nekonečné smyčce vrtí ocáskem a mrká.

Část o internetu je zakončena vysvělením principu vyhledávání na internetu spojeném s upozorněním, že ne všechny informace, které lze najít na internetu, jsou správné nebo pravdivé. Ilustrační obraz představuje stylizovanou vyhledávací lištu prohlížeče a Ajtáčka s lupou (symbol hledání).



Obrázek 6.34: Interaktivní informační grafika o zkratkách států na internetu.

6.8.9 Kvíz o internetu

Kvíz o internetu je vytvořený na stejném principu jako kvíz o bezpečnosti s počítačem. Jediný rozdíl je zde v datovém souboru JSON s příslušnými otázkami a obrázky a s pozměněnými komentáři o výsledku kvízu.



Obrázek 6.35: Ukázka zpětné vazby při špatné odpovědi v kvízu.

6.8.10 O šifrách

Tato kapitola se zabývá principiálním vysvětlením základu a významu šifrování a kódování. Textový úvod se zmiňuje jak o starých šifrách ze starého Říma, tak o současných kódech, se kterými je možné se setkávat každý den, jako jsou například QR a čárové kódy (tyto kódy ilustruje příslušný obrázek opět s maskoty).

Následuje interaktivní grafika vytvořená s pomocí rozhraní Canvas. Opět na interaktivitu upozorňuje svou animací myška, která v textové bublině komentuje možnosti interakce. Grafika představuje ukázkou jednoduchého kódování barev pomocí počátečního písmene barev.

Závěrečná část představuje koncept elektronického podpisu na základní úrovni tak, aby tento koncept mohly pochopit i děti. Text doplňuje obrázek Ajťáčka s klíčem (který představuje elektronický podpis) a otevřeného dokumentu na počítači.

Můžeme zakódovat i obrázek!
V tabulce je nakreslený křížek.
Každá barva má svůj kód.
Zkus zjistit, jaké barvy se skrývají
pod kódy v políčkách tabulky!

M	Ž	M
Ž	Ž	Ž
M	Ž	M

Obrázek 6.36: Interaktivní grafika představující jednoduché zakódování obrázku.

6.8.11 Šifrovací hra

Hra ke kapitole o kódech dětem přináší úkol vyřešit šifru, která spočívá v sestavení věty pomocí jejích fragmentů seřazených v tabulce. Samotná šifra je reprezentována uspořádanými dvojicemi čísel, která označují danou buňku v tabulce, kde první číslo představuje řádek a druhé sloupec v tabulce. Fragmenty věty se poté musí přesunout ve správném pořadí do šedých obdélníků.

Při kliknutí na přetahnutelný objekt, v tomto případě obdélník s částí výsledné věty, by se měla dostavit zpětná vazba o tom, že opravdu lze s daným objektem nějak manipulovat. Když tedy uživatel klikne na některý obdélník, sníží se jeho viditelnost. Pod každou buňkou tabulky se nachází jejich kopie, aby tabulka pro rozluštění šifry byla stále k dispozici.

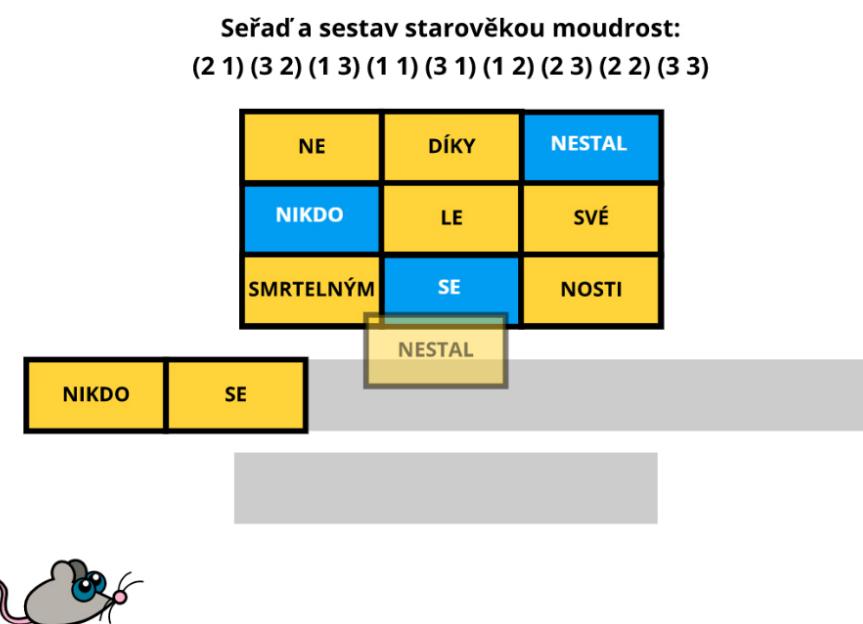
Místo, kam se mají prvky přesunout, by mělo být jasně označeno. V případě šifrovací hry je označeno šedou plochou (myška po najetí kurzoru také připomíná cílovou oblast pro buňky tabulky). Při přetahnutí objektu na správné místo by se měla projevit adekvátní zpětná vazba, v tomto případě, že blok „zapadne“ do svého správného přesného místa, když je fragment věty upuštěn v dostatečné blízkosti cílového místa. [6]

Přesouvání je realizováno pomocí metody `on()` knihovny EaselJS, která naslouchá při daném objektu dané události, v tomto případě stisknutí myši, pohyb myši a při uvolnění tlačítka myši. Při uvolnění tlačítka se vždy provede test, je-li daná část dostatečně blízko svému cílovému místu a jsou-li jsou již všechny části na svém místě.

```
if (piece) {
    piece.name = pieceName;
    piece.on("mousedown", function (evt) {
        this.alpha = 0.5;
        this.parent.addChild(this);
        this.offset = {
            x: this.x - evt.stageX,
            y: this.y - evt.stageY
        };
    });
    piece.on("pressmove", function (evt) {
        this.x = evt.stageX + this.offset.x;
        this.y = evt.stageY + this.offset.y;
    });
    piece.on("pressup", function (evt) {
        this.alpha = 1;
        var target = this.parent["t" + this.name.substr(1)];
        if (target && hitTestInRange(target, 50)) {
            this.x = target.x;
            this.y = target.y;
        }
        completedTest();
    });
}
```

Obrázek 6.37: Realizace přetažení jednotlivých částí věty.

Po úspěšném vyřešení se objeví dialog s veselou myškou a opět s povzbuzením a radou na prozkoumání dalších kapitol.



Obrázek 6.38: Ukázka tahu fragmentem věty v Šifrovací hře.

7 Uživatelské testování se žáky 1. stupně

Při testování použitelnosti softwaru je nutné brát na zřetel, že děti jako uživatelé se velmi liší od dospělých. Tomu je třeba se také přizpůsobit.

7.1 Obecná doporučení při uživatelském testování s děmi

Děti navštěvující základní školu lze zapojit do testování použitelnosti bez větších problémů. Díky školní docházce dokáží pracovat vsedě na úkolu a následovat instrukce od dospělého. Nejsou si většinou ani vědomy, že je někdo při práci s počítačem pozoruje. Dokáží bez problému zodpovídat dotazy a orientovat se v nových situacích. V tomto věku si také vytváří schopnost sofistikovaně popisovat objekty, které vidí, a činnosti, které dělají. V drtivé většině již mají zkušenosti s počítačem a jsou připravené kritizovat testovaný software určený pro děti.

Celkové testování by mělo trvat maximálně 45 minut, po této době jsou děti unavené a ztrácí pozornost. Vedení dětí a jejich koordinace může být pro testujícího velmi náročné, a proto je dobré dělat mezi testováním přestávky. [41]

Pokud dítě testujícího nezná, je na začátku testování vhodné navázat s dítětem neformální kontakt. Děti většinou rády mluví o svých zážitcích, oblíbených počítačových hrách apod. Je důležité je upozornit na to, že nejsou testovány ony samotné, ale testován je daný software. Pocit důležitosti jim může dodat fakt, že testování může být „přísně tajné“. Je dobré je také připravit, co je během testování nebo v samotné aplikaci čeká. Pokud se cítí s cizími lidmi nepříjemně, je vhodné mít na blízku jím známou osobu nebo rodiče, kteří ale nesmí zasahovat do průběhu testu. Je samozrejmostí, že v průběhu testu nesmí děti rozptylovat různé ruchy a zvuky nebo samotné prostředí. [41]

Od dětí mladšího školního věku je možné během testu očekávat, že jsou schopné plnit specifické úkoly. Je také možno je nechat volně procházet produkt. Děti jsou zvyklé při zkoušení s novými aplikacemi nebo hrami být s dalšími lidmi – at' už jsou to jejich rodiče, sourozenci nebo kamarádi. Jsou navyklé se ptát, když si nejsou jisté, co mají dělat. Je nutné jim neodpovídat přímo, ale spíš návodnou otázkou. Nevhodné je pokládat otázku, jestli si chtejí vyzkoušet daný software, protože to jim dává možnost odmítnout. Při práci je důležité děti podporovat a poskytovat jim pozitivní zpětnou vazbu. [41]

Při vyhodnocování je třeba zahrnout i to, jak se děti tvářily nebo reagovaly (úsměv, smích, zvýšená pozornost, vzdechy, zívání) při testování aplikace. Tyto reakce jsou někdy spolehlivější než odpovědi samotných dětí. Po skončení testování je nutné děti pochválit a poděkovat jim jak verbálně, tak i drobným dárkem. [41]

7.2 Testování interaktivní učebnice Ajťáček

7.2.1 Částečné testování

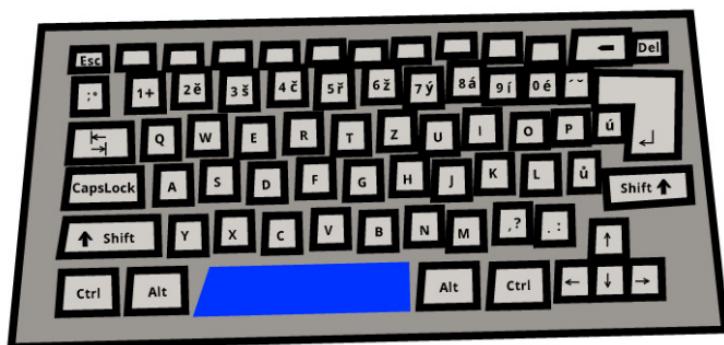
Částečné neformální testování probíhalo během vývoje aplikace pomocí metody paper and pencil, tedy tak, že byl testovaný subjekt sledován při plnění úloh a testující si dělal poznámky. Tato metoda je rychlá a levná, ale bohužel zachytí jen ty nejzásadnější chyby v návrhu. [42]

Tohoto testování se zúčastnilo 5 dětí (3 chlapci, 2 dívky) ve věku 10–11 let navštěvující 4.–5. třídu 1. stupně základní školy. Jediné zachycené problémy byly s návrhem menu, kdy si děti stěžovaly na neustálou nutnost scrollování, protože menu zabíralo velkou část obrazovky. Problém byl vyřešen nízkou lištou s fixní pozicí s menší variantou loga a tlačítka s piktogramy.



Obrázek 7.1: Ukázka upraveného menu.

Dalším problémem bylo přehlédnutí interaktivity klávesnice v kapitole Píseme. Tento problém byl vyřešen blíkajícími klávesami v době před započetím práce s grafikou klávesnice.



Obrázek 7.2: Některé z kláves na klávesnici blikají pro upoutání pozornosti.

Posledním problémem bylo nastavení obtížnosti u Programovací hry, kde byl navýšen počet z původních 25 možných příkazů na 35 a byla změněna pozice některých písmen. Děti tak mohou hru lépe zvládat, ale přesto je stále ještě pro ně výzvou.

7.2.2 Celkové testování

Celkového testování se zúčastnilo 10 dětí (6 dívek a 4 chlapci) ve věku 9–11 let navštěvující 3.–5. třídu 1. stupně základní školy. Výběr účastníků probíhal tedy podle cílové skupiny uživatelů učebnice. Všechny děti měly zkušenosti s prací s počítačem na každodenní bázi. Dvě z testovaných dětí ještě neměly na škole předmět o informatici nebo o informačních technologiích, což se ale neprojevilo v jejich průchodu aplikací ani v jejich hodnocení. Každé testování trvalo v průměru přibližně 30 minut, děti byly v klidném prostředí (pokud prostředí neznaly, bylo jim dovoleno si místo prohlédnout a povídат si o něm). Jestliže děti testujícího neznaly, byla vždy na blízku známá osoba (ve většině případů jeden z rodičů) a testující se jim představil.

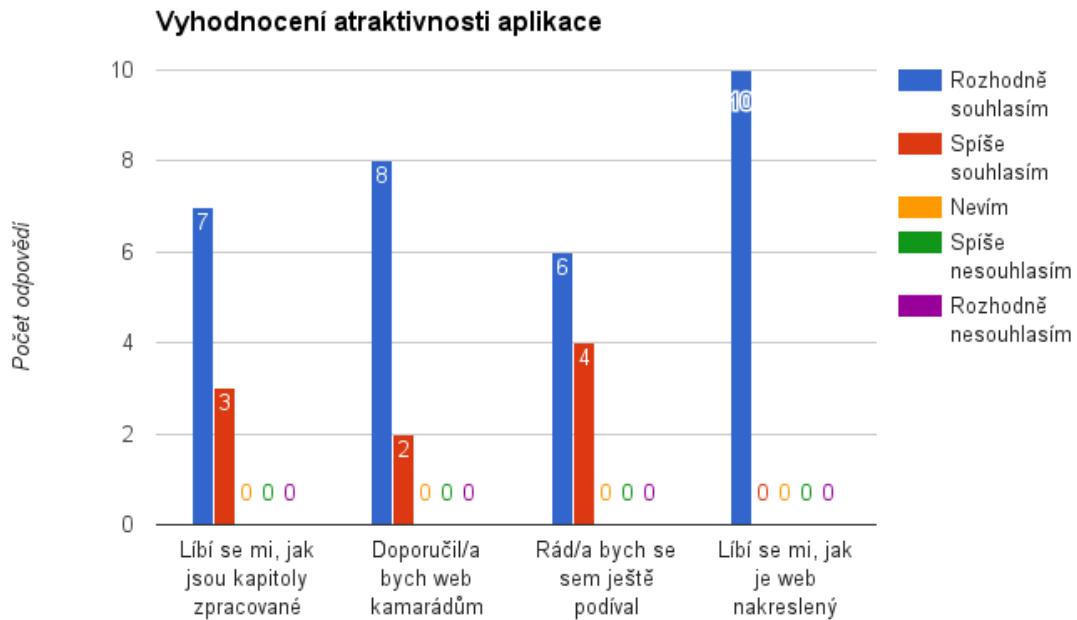
Na začátku testování proběhl krátký nezávazný rozhovor, který se postupně přenesl na otázky na výuku informatiky ve škole, zkušenosti s počítačem apod. Poté byla představena aplikace a důvod, proč chlapec nebo dívka byly k testování přizvání. Děti byly upozorněny na svou důležitost při testování. Nejdříve probíhalo testování metodou paper and pencil. Děti dostaly úkol projít volně aplikace podle svého tempa a roz hodnutí a byly povzbuzeny ke komentování svých akcí nebo kritizování aplikace při své práci. Pokud přehlédly nějakou část nebo prvek, byly po konci průchodu aplikací nasměrovány k témtoto prvkům.

Po testování následoval rozhovor o aplikaci a její kritika (co se nového dozvěděli z aplikace, co je bavilo, co jim vadilo nebo jestli by neměly samy nějaké nápady na zlepšení). Poté dostaly krátký dotazník. Jeho jednotlivé části jim byly vysvětleny testujícím. Dotazník obsahoval údaje o věku a třídě testovaného dítěte a čtyři tvrzení o atraktivnosti aplikace, který děti hodnotily podle Likertovy škály (Líbí se mi, jak jsou kapitoly zpracované; Doporučil/a bych web kamarádům; Rád/a bych se sem ještě podíval/a; Líbí se mi, jak je web nakreslený). Dotazník byl zakončen dvěma otevřenými otázkami – co se dětem v aplikaci nejvíce líbilo a co se jim naopak nelíbilo. Na rozloučenou každý účastník dostal placku s maskotem učebnice Ajťáčkem.

7.2.3 Výsledky celkového testování

Všechny děti aplikace zaujala a komentovaly ji pozitivně. Nejvíce děti nadchly animace, u nichž měly možnost manipulace. Zaujal je také styl animace grafiky a maskotů. Jak je vidět v grafu 7.1, nejméně si děti byly jisté s návratem na stránku, naopak všechny byly zcela přesvědčené o atraktivnosti grafiky.

Sedm z testovaných dětí uvedlo v dotazníku jako nejoblíbenější část aplikace kapitolu Programujeme a Programovací hru. Přestože některým účastníkům trvalo delší dobu hrů dokončit (cca 15 minut), již při testování ji velmi chválily. Dvě z testovaných dětí ji komentovaly jako až příliš těžkou, ale vizuálně přitažlivou. Je nutné však dodat, že tyto dva subjekty jako jediné nehrájí často nebo nehrájí vůbec počítačové hry (zjištěno z rozhovoru) a tento faktor by mohl být vysvětlením jejich názoru. Ostatní děti označily za nejlepší část kapitoly O šifrách (2 děti) a kvízy a kapitolu Píšeme (jeden účastník).



Graf 7.1: Vyhodnocení sběru dat z dotazníků.

U otevřené otázky o negativech aplikace drtivá většina dětí nevěděla, co napsat, ani s pomocí testujícího, který je upozornil na obtížnost Programovací hry, Šifrovací hry apod. Pouze jeden účastník zmínil přání o zařazení více her do aplikace.

Při testování děti neztrácely pozornost a nevykazovaly žádné nonverbální negativní reakce. Nejvíce zvýšená pozornost byla zaznamenána u interaktivní klávesnice, Psací hry, animace o programování, Programovací hry, animace o bezpečnosti a Šifrovací hry.

Každý účastník testování procházel kapitoly učebnice v jiném pořadí, což v tomto případě nehraje roli, jelikož jednotlivé kapitoly nevyžadují znalosti některé z ostatních. Většina účastníků však přehlížela animované postavičky myšek poskytujících nápovědu v Programovací a Šifrovací hře a nevyužila ji. Tuto nápovědu ale děti evidentně nepotřebovaly.



Obrázek 7.3: Odměna pro děti, které se zúčastnily testování – placka s Ajtáčkem.

Závěr

Cílem mé práce bylo vytvořit návrh elektronické interaktivní učebnici informatiky a programování ve formě responzivní webové aplikace se zaměřením na žáky 1. stupně základní školy. Jsem si vědom, že nemám pedagogické vzdělání, to jsem se však snažil si doplnit studiem odborné literatury. Textový a didaktický obsah jsem z drtivé většiny převzal z učebnice „Informatika pro 1. stupeň základní školy“ od Jiřího Vaníčka. Backend webové aplikace je vytvořen pomocí softwarového systému Node.js a jeho frameworku Express. Aplikaci tvoří pět kapitol o psaní na klávesnici, o bezpečnosti práce s počítačem, o základech programování, o internetu a o šifrách. Každá z těchto částí obsahuje interaktivní informační grafiku vytvořenou s pomocí HTML5 Canvas a programu Adobe Flash Professional CC, kterou děti mohou projít svým vlastním tempem. Tuto grafiku doprovází také mnoho autorských ilustrací. Ke každé kapitole je připojena jednoduchá hra nebo kvíz, kde bylo opět použito rozhraní Canvas nebo javascriptová knihovna jQuery. Celá aplikace je provedena v jednotném vizuálním stylu, ve kterém figuruje hlavní maskot Ajťáček a jeho pomocníce myšky. Výsledná aplikace je k dispozici na adrese <http://ajtacek-0491.rostiapp.cz/> nebo na přiloženém DVD v jeho offline verzi.

Při práci jsem používal nejen své vlastní zkušenosti při vytváření grafiky pro děti, ale čerpal jsem také z odborných zdrojů na toto téma. Tato práce tedy obsahuje souhrn znalostí jak teoretických, tak praktických, které jsou třeba pro vytvoření podobné webové aplikace. Zde vidím hlavní přínosem této diplomové práce. Dalším mým přínosem je samotná aplikace Ajťáček, na které je možné pracovat dále a vytvořit kompletní moderní výukový materiál. Je také možné pracovat dál na této aplikaci a přidat jí další funkce, např. vytvořit jednoduchý informační systém s přihlašováním a hodnocením jednotlivých uživatelů apod.

Aplikace je ale i v současné podobě schopná předávat znalosti určitých oblastí z informatiky a pomocí uživateli osvojit si je.

Na mou práci se dá z pedagogického hlediska navázat právě vytvořením didaktického obsahu přímo pro interaktivní učebnici. Je také možné vytvořit další interaktivní učebnice pro jiné obory. Tyto učebnice lze také přetvořit jako aplikace pro mobilní zařízení a využít jejich možnosti interakce.

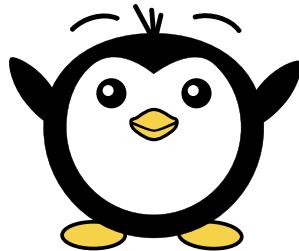
Použité zdroje

- [1] Zounek, Jiří; Šeďová, Klára. *Učitelé a technologie: mezi tradičním a moderním pojetím*. 1. vyd. Brno: Paido, 2009, 172 s. ISBN 9788073151874.
- [2] Zounek, Jiří. *E-learning – jedna z podob učení v moderní společnosti*. Brno: Masarykova univerzita, 2009, 161 s. ISBN 9788021051232.
- [3] Zounek, Jiří. *ICT v životě základních škol*. Praha: Triton, 2006, 151 s. ISBN 8072548581
- [4] Metodický portál RVP.CZ. *Vymezení Rámcového vzdělávacího programu pro základní vzdělávání v systému kurikulárních dokumentů* [online]. [cit. 2015-09-13]. Dostupné z: <http://digifolio.rvp.cz/view/view.php?id=6443/>.
- [5] Ministerstvo školství, mládeže a tělovýchovy. *Rámcový vzdělávací program pro základní vzdělávání* [online]. [cit. 2015-09-13]. Dostupné z: http://www.msmt.cz/uploads/soubory/zakladni/SP_RVPZV_2007.zip/.
- [6] Cooper, Alan; Reimann, Robert; Cronin Dave. *Aboutface 3: the essentials of interaction design*. Indianapolis: Wiley, 2007, 610 s. ISBN 9780470084113.
- [7] Graham, Lisa. *Principles of Interactive Design*. 1. vyd. Independence: Delmar Cengage Learning, 1998, 240 s. ISBN 0827385579.
- [8] Grannell, Craig. *The graphic designer's guide to interactive design* [online]. [cit. 2015-10-02]. Dostupné z: <http://www.digitalartsonline.co.uk/features/creative-lifestyle/graphic-designers-guide-interactive-design/>.
- [9] Rogers, Simon. *Infographics for children: what they can learn from data visualisations* [online]. [cit. 2015-10-09]. Dostupné z: <http://www.theguardian.com/news/datablog/2014/mar/07/infographics-for-children-can-learn-from-data-visualisations/>.
- [10] Bers, Maria U. *Designing for Children: Supporting Positive Youth Development through Social Media* [online]. [cit. 2015-06-25]. Dostupné z: <http://uxpamagazine.org/designing-for-children-supporting-positive-youth-development-through-social-media/>.
- [11] Nielsen, Jakob. *Children's Websites: Usability Issues in Designing for Kids* [online]. [cit. 2015-04-11]. Dostupné z: <http://www.nngroup.com/articles/childrens-websites-usability-issues/>.
- [12] UX Magazine. *Design for Experience: Experience for Children* [online]. [cit. 2015-05-13]. Dostupné z: <http://uxmag.com/articles/design-for-experience-experience-for-children/>.

- [13] Fishel, Catharine M. *Designing for children: marketing design that speaks to kids*. 1. vyd. Gloucester: Rockport, 2001, 160 s. ISBN 9781564968005.
- [14] Doherty, Rina. *Looking Closely at e-Learning: Vision Research Reveals Ways to Improve Children's Experiences* [online]. [cit. 2015-07-20]. Dostupné z: http://uxpamagazine.org/looking_closely_elearning/.
- [15] Idler, Sabina. *5 Key Difference Between Kids And Adults* [online]. [cit. 2015-08-02]. Dostupné z: <http://uxkids.com/blog/5-key-difference-between-kids-and-adults/>.
- [16] Morrison, David. *Best Practices For Web Design For Kids* [online]. [cit. 2015-10-15]. Dostupné z: <http://www.smashingmagazine.com/2011/07/best-practices-for-designing-websites-for-kids/>.
- [17] Khan Academy. *Our mission is to provide a free, world-class education for anyone, anywhere*. [online]. [cit. 2015-03-14]. Dostupné z: <https://www.khanacademy.org/about/>.
- [18] Khanova škola. [online]. [cit. 2015-11-25]. Dostupné z: <https://khanovaskola.cz/>.
- [19] Moodle. *Moodle 2.6 documentation* [online]. [cit. 2015-10-04]. Dostupné z: <https://docs.moodle.org/26/en/>.
- [20] Stehlíková, Naďa; Cachová, Jana. *Konstruktivistické přístupy k vyučování a praxe* [online]. [cit. 2015-09-23]. Dostupné z: <http://class.pedf.cuni.cz/NewSUMA/FileDownload.aspx?FileID=89>.
- [21] Metodický portál RVP.CZ. *Masivní otevřené online kurzy*. [online]. [cit. 2015-03-02]. Dostupné z: <http://spomocnik.rvp.cz/clanek/10725/>.
- [22] Pappano, Laura. *The Year of the MOOC*. [online]. [cit. 2015-03-02]. Dostupné z: http://www.nytimes.com/2012/11/04/education/edlife/massive-open-online-courses-are-multiplying-at-a-rapid-pace.html?pagewanted=all&_r=2>.
- [23] Tynker. [online]. [cit. 2015-10-31]. Dostupné z: <https://www.tynker.com/>.
- [24] Hour of code. *Budování galaxie s kódem* [online]. [cit. 2015-11-20]. Dostupné z: <https://studio.code.org/s/starwarsblocks/stage/1/puzzle/1/>.
- [25] Mohan, Surendra. *Node.js Essentials*. Birmingham: Packt Publishing, 2014, 196 s. ISBN 9781783987511.
- [26] Erb, Benjamin; Müller, Michael. *node.js: Single-threaded and event-driven JavaScript outside your browser* [online]. [cit. 2015-05-24]. Dostupné z: <http://berb.github.io/nodejs-talk-chaosseminar/#1/>.
- [27] Spolsky, Joel. *Human Task Switches Considered Harmful* [online]. [cit. 2015-03-10]. Dostupné z: [http://www.joelonsoftware.com/articles/fog0000000022.html/](http://www.joelonsoftware.com/articles/fog0000000022.html).

- [28] Express.js [online]. [cit. 2015-08-13]. Dostupné z: <http://expressjs.com/>.
- [29] Bray, Tim. *JSON Redux AKA RFC7159* [online]. [cit. 2015-09-10]. Dostupné z: <https://www.tbray.org/ongoing/When/201x/2014/03/05/RFC7159-JSON/>.
- [30] Benedetti, Ryan; Cranley, Ronan. *Head First jQuery*. Newton: O'Reilly Media, 2011, 540 s. ISBN 9781449393212.
- [31] Lubbers, Peter; Albers, Brian; Salim, Frank. *HTML5: programujeme moderní webové aplikace*. Vyd. 1. Brno: Computer Press, 2011, 304 s. ISBN 9788025135396.
- [32] Boulton, Jim. *Macromedia Flash Back* [online]. [cit. 2015-11-02]. Dostupné z: <http://digital-archaeology.org/flash-back/>.
- [33] Adobe Systems. *Nápořeď k aplikaci Adobe Flash Professional CC* [online]. [cit. 2015-11-30]. Dostupné z: https://helpx.adobe.com/cz/pdf/flash_reference.pdf.
- [34] Lee, Rich. *Welcome Adobe Animate CC, a new era for Flash Professional* [online]. [cit. 2015-12-10]. Dostupné z: <http://blogs.adobe.com/flashpro/welcome-adobe-animate-cc-a-new-era-for-flash-professional/>.
- [35] CreateJS. [online]. [cit. 2015-10-05]. Dostupné z: <http://www.createjs.com/>.
- [36] Marcotte, Ethan. *Responsive Web Design*. A Book Apart, 2011. 153 s. ISBN 9780984442577.
- [37] Lupton, Ellen. *Type on screen: a guide for designers, developers, writers and students*. New York: Princeton Architectural Press, 2014, 208 s. ISBN 9781616891701.
- [38] Dostál, Martin. *Dotyková rozhraní*. Brno, 2015. Přednáška. Masarykova univerzita.
- [39] Vaníček, Jiří. *Informatika pro 1. stupeň základní školy*. 1. vyd. Brno: Computer Press, 2012, 88 s. ISBN 9788025137499.
- [40] Jochmann-Mannak, Hanna; Lentz, Leo; Huibers, Theo; Sanders, Ted. *Three Types of Children's Informational Web Sites: An Inventory of Design Conventions*. [online]. [cit. 2015-10-02] Dostupné na http://eprints.eemcs.utwente.nl/22641/01/3Jochmann_TechComm_Nov4thQrt_WEB.pdf.
- [41] Hanna, Libby; Risden, Kirsten; Alexander, Kirsten. *Guidelines for Usability Testing with Children*. [online]. [cit. 2015-12-01] Dostupné na <http://www.microsoft.com/usability/UEPostings/p9-hanna.pdf>.
- [42] Dostál, Martin. *Observační metody*. Brno, 2015. Přednáška. Masarykova univerzita.

Obrazové přílohy



AJŤÁČEK

Kolik ti je let?

Do které třídy chodíš?

	Rozhodně souhlasím	Spíše souhlasím	Nevím	Spíše nesouhlasím	Rozhodně nesouhlasím
Líbí se mi, jak jsou kapitoly zpracované					
Doporučil/a bych web kamarádům					
Rád/a bych se sem ještě podíval/a					
Líbí se mi, jak je web nakreslený					

Co se ti u Ajtáčka líbilo?

Co se ti u Ajtáčka nelíbilo?
Co by si změnil/a?

• AJŽÁČEK

A PÍSEME B BEZPEČNĚ P PROGRAMOVANÉ O INTERNETU O ŠIFRÁCH

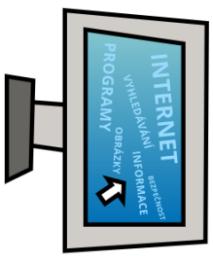
VÍTEJ U AJŽÁČKA

Vdej se s tučňákem Ajžáčkem na cestu poznání, kde se dozvídáš spoustu věcí o počítači a jeho tajemstvích. Čeká tě mnoho obrázků, animací, her a vědomostí, díky kterým budelé moci ovládat počítač a jeho programy a vybavení, a také bezpečně „brouzdat“ nebo vyhledávat na internetu.

Neváhej a klikni na některou z kapitol v nabídce (Píseme, Bezpečně, Programovanie, O internetu nebo O šifrách). Každá z kapitol obsahuje v sobě kvíz nebo hru, kterou si můžeš zahrát! Aby si ale hru zvládl/a, musíš si napřed projít danou kapitolu.

Seznam se!

Na těchto stránkách tě bude provázek šikovný tučňáček Ajžáček a jeho pomocníci myšky. Kdykoliv uvidíš, že na tebe Ajžáček nebo myšky mrkají, určitě ti chtejí něco říct! Můžeš to zjistit, pokud na ně najedeš myší.



TUČŇÁČEK
AJŽÁČEK



MYŠKY

PÍSEME NA KLÁVESNICI

Na klávesnici najdeme nejen písmena a číslice, ale i jiné znaky, například tečky a čárky ve větě, otazník a vykříčník, závorky, znaménka pro počítání a jiné.

Při tukání na klávesy s písmeny se piší pouze malá písmena. Abychom napsali velká, musíme současně držet stisknutou klávesu **Shift** nebo stisknout klávesu **Caps Lock**, která přepíná mezi malými a velkými písmeny.

Některá písmena mají nad sebou háček, háček nebo kroužek. Většinu takových písmen najdeme na klávesnici.

Mnoho tlačítek je na všech klávesnicích na stejném místě (například písmena, mezerník nebo klávesa enter). Existuje ale více modelů klávesnice a některé se liší v umístěních znaků nebo tlačítek pro funkce.



PROZKOUMEJ TLAČÍTKA
NA KLÁVESNICI!!



Jak ale napišeme písmena s háčky a čárkami, která na klávesnici nejsou?

Takové písmeno se musí napsat nadvákrát. Nejprve se stiskne klávesa s háčkem a čárkou (levá od tlačítka pro mažání). Chceme-li napsat háček, musíme držet klávesu **Shift**, potom napišeme písmeno.

Nepříjemné je, že po napsání háčku nebo čárky není nic vidět a také že nejprve musíme napsat znaménko a pak teprve písmeno. Po tréninku si ale jistě zvykneš.

$$\begin{array}{l} \text{~} + \text{~} = \acute{a} \\ \text{\uparrow Shift} \quad \text{\uparrow} + \text{~} = \check{s} \\ \text{\uparrow} + \text{\uparrow Shift} \quad \text{~} + \text{\uparrow Shift} \quad \text{~} = \acute{A} \\ \text{\uparrow Shift} \quad \text{\uparrow} + \text{\uparrow Shift} \quad \text{~} = \check{S} \end{array}$$



Vyzkoušej Psací hru a najdi za minutu co nejvíce písmenek na klávesnici.

PSACÍ HRA

O aplikaci

Copyright © Jakub Špířík

PSACÍ HRA

Máš minutu na to, aby si napsal/a co nejvíce písmenek. Hledej na klávesnici!

Nezapomeň u velkých písmen přidržet klávesu **Shift**!

START

BUDĚŠ MÍT NA
TO 60 SEKUND!



O aplikaci

Copyright © Jakub Špířík

BEZPEČNĚ S POČÍTAČEM

S počítačem patrně strávíš velkou část života, daleko více než tvoji rodiče. Proto je třeba se u něj umět správně a bezpečně chovat. Je důležité vědět, jak chránit své zdraví i zdraví počítače.



Stisknutím šipky
spusť animaci



Kde je vhodné místo pro počítač

Když je počítač umístěn u okna, světlo z okna oslnuje nebo monitor není dobrě vidět.

Počítač naproti oknu chytá odlesky zvenku, což je nepříjemné.

Nejlepší místo pro počítač může být dál od okna a nebo může také zatahnout závěs. Tím se ale v pokoji setmí, i když je den, a to také není dobré.



Vyzkoušej své vědomosti v kvízu o bezpečnosti práce s počítačem!

KVÍZ O BEZPEČNOSTI

O aplikaci

Copyright © Jakub Špířík

KVÍZ O BEZPEČNOSTI S POČÍTAČEM

Kde je vhodné místo pro počítač?



U okna

Dále od okna

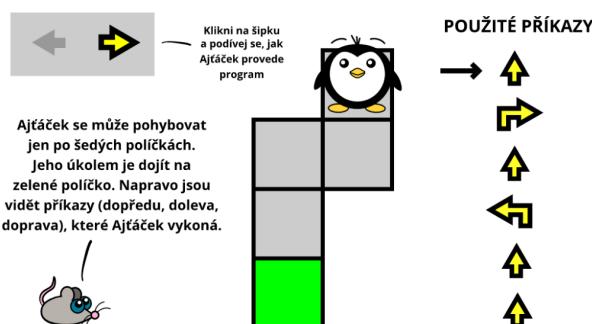
Naproti oknu

O aplikaci

Copyright © Jakub Špířík

PROGRAMUJEME

Každý program, každou počítačovou hru musel někdo vytvořit, naprogramovat. My si v této kapitole něco takového vyzkoušíme. Pomoci příkazů si ukážeme, jak ovládat postavíčku tučňáka Ajťáčka na obrazovce. Když program vynutíme správně, bude nás Ajťáček poslouchat a dělat to, co chceme. Když v programu uděláme chybu, bude se chovat jinak, než chceme.



Pomocí programu se dá zapsat postup, tedy co se udělalo nejdříve a co potom. Může se tak zaznamenat, jak vyrobit nějaký výrobek (výrobní postup), jak připravit jídlo (kuchařský recept), jak někam dojet (navigace) nebo zahrát hudební dílo (notový zápis).

Takový pracovní postup může být klidně i návod na to, jak uvařit čaj:

DO PRÁZDNÉHO HRNKU VLOŽ PYTLÍK S ČAJEM,
DO RYCHLOVARNÉ KONVICE NALEJ VODU,
ZAPOJ KONVICI DO ELEKTRINY A ZAPNÍ JI,
POČKEJ 5 MINUT,
NALEJ VODU Z KONVICE DO HRNKU,
VYNDEJ PYTLÍK Z HRNKU,
ČAJ JE HOTOV



Pomoz Ajťáčkovi splnit úkol při sbírání písmenek!

PROGRAMOVACÍ HRA

O aplikaci

Copyright © Jakub Špířík

PROGRAMOVACÍ HRA



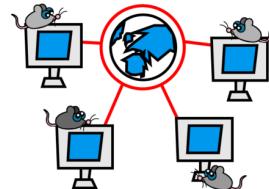
Zbývá příkazů: 35

O aplikaci

Copyright © Jakub Špířík

O INTERNETU

Co byly počítače bez internetu! Internet propojuje počítače tak, aby mohly posílat obrázky, zprávy, články nebo videa z jednoho konce světa na druhý. K internetu jsou připojeny stovky milionů počítačů nebo třeba mobilních telefonů a tabletů. Když jste připojeni k internetu, máme možnost podívat se do jiného počítače a stahnout si z něj, co nám nabízí a co nás zajímá.



K internetu se můžeš dostat přes webový prohlížeč stránek, který i teď máš otevřený. Umožňuje zobrazovat stránky, které jsou umístěny na jiných počítačích, které jsou připojeny k internetu. Prohlížení internetu není totéž jako práce v jiných aplikacích. Někdy to může trvat dlouho, než se stránka zobrazí, pokud například obsahuje velké množství obrázků.

Internetová adresa

Aby se počítače připojené k internetu neoplopyly, každý dostal svoje jedinečné jméno, kterému říkáme adresa. Ta se skládá z několika slov oddělených tečkami. Podle této adresy se pozná například, v jaké zemi se počítač nachází.

<http://www.ceskaskola.cz/archa/navod.htm>



Internetové stránky mají na konci své adresy připsány zkratky, podle kterých se dá poznat, z jaké země stránka pochází.
Zjisti, jaké mají zkratky státy kolem Česka!



Hledáme na internetu

Je snadné se dostat na stránku, když víme, jak se jmenuje. Jak ale zjistit, na které stránce je povídáno o tučňácích, skautech, o sopkách, o oblíbeném hokejovém klubu nebo zpěvákoví? K tomu slouží vyhledávání.

 tučňáci

Když do řádku adresy napišeme něco jiného než adresu, můsto webový stránky se zobrazí seznam, v němž jsou vypsány odkazy, které počítač vyhledal a souvisejí s napsaným slovem. Pak se můžeme vybrat, na který odkaz klikneme a kterou stránku ze seznamu si prohlédneme. Místo psaní do řádku adresy můžeme napsat stránku vyhledávače, například google.cz (čti gугл).

Vyhledávací řádek se zobrazí uprostřed stránky.

Ale pozor! To, co čteme na internetu, nemusí být vždycky pravda, protože na internet může psát každý a může lhát!



Vyzkoušej své vědomosti v kvízu o internetu!

KVÍZ O INTERNETU

O aplikaci

Copyright © Jakub Špírk

KVÍZ O INTERNETU

Jaká zkratka se používá v Internetových adresách pro Českou republiku?



.CR

.CZ

.CS

O aplikaci
Copyright © Jakub Špiřík.

O ŠIFRÁCH

Lidé, kteří si posílají e-maily, by nebyli rádi, kdyby si jejich poštu mohli číst kdokoliv jiný. E-mail není zlepšený v obálce, spíše připomíná pohlednice, kterou si mohou na poště předčít. Proto se e-maily, ale i jiné důvěrné informace šifrují. Tajný kód potom přečte pouze ten, kdo zná způsob, jak šifru rozluštít. Poštovní aplikace šifrují e-mailové zprávy automaticky.

Šifrování zpráv se používalo odédavna. Například římský imperátor César před více než 2 000 lety používal pro vojenské zprávy takovouto šífru: všechna písmena posunul v abecedě o tři místa. Místo písmenka A se napsalo D, písmeno R se zašifrovalo jako U (háčky a čárky Římané nepoužívali). Potom bylo takové slovo ROMA zašifrováno URPD. Správný adresát šífru znal, posunul písmena abecedy o tři místa dopředu a rozluští ji.

V předminulém století se pro dálkové posílání zpráv používal telegraf. Dokázal po dlouhých drátech přenést dlouhé nebo krátké písmeno. Pro kódování písmen se pak používala Morseova abeceda, v níž každé písmeno představovalo několik teček a čárk (krátkých a dlouhých písknutí).

V současné době se hodně používají čárové kódy nebo čtvercový kód (někdy označovaný QR kód). Najdeme je třeba na obalech výrobků. Když čteci zařízení přečte kód, počítá pozná, o jaký výrobek se jedná.



Můžeme zakódovat i obrázek!
V tabulce je nakreslený křízek.
Každá barva má svůj kód.
Zkus zjistit, jaké barvy se skrývají
pod kódy v políčkách tabulky!



M	Ž	M
Ž	Ž	Ž
M	Ž	M

Co je to elektronický podpis?

Elektronický podpis je tajný kód (říká se mu klíč), podle kterého lze ověřit, že nějaký dokument vytvořila určitá osoba, úřad nebo firma. Pokud bys vlastnil elektronický podpis, mohl bys tímto tajným kódem zašifrovat text napsaný v počítači. Ten pak přuje přečíst a každý si bude moci ověřit, že tento dokument je pravý, nezfašovaný. Elektronický podpis je důležitý pro ověřování obchodních smluv, soudních a policejních materiálů a dalších elektronických dokumentů.



Vyzkoušej si využítit šifru!

ŠIFROVACÍ HRA

O aplikaci

Copyright © Jakub Špirík

ŠIFROVACÍ HRA

V této šíffře je popsáno, v jakém pořadí čistit buňky tabulky. Dvojice (2 3) znamená druhý řádek, třetí sloupec.

Poskládej slova nebo jejich části do šedých obdélníků.

Seřaď a sestav starověkou moudrost:

(2 1) (3 2) (1 3) (1 1) (3 1) (1 2) (2 3) (2 2) (3 3)

NE	DÍKY	NESTAL
NIKDO	LE	SVÉ
SMRTELNÝM	SE	NOSTI



O aplikaci

Copyright © Jakub Špirík

O APLIKACI

Elektronická interaktivní učebnice Ajťáček vznikla jako diplomová práce na Fakultě informatiky Masarykovy univerzity v Brně.

Tato aplikace slouží jako ukázka použití technik vytváření grafiky a animací pro tvorbu učebního materiálu a ne jako celistvá učebnice.

KONTAKT

Jakub Špirík

jakub.spirik@gmail.com

O aplikaci

Copyright © Jakub Špirík

Obsah přiloženého DVD

Přiložené DVD obsahuje:

- kompletní text diplomové ve formátu .pdf.
- zdrojové kódy webové aplikace Ajtáček.
- offline verzi webové aplikace Ajtáček.
- zdrojové soubory animací vytvořených s pomocí rozhraní Canvas ve formátu .fla.
- dotazník ve formátu .pdf použitý při testování aplikace.