

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta elektrotechniky a informatiky

Evidenčné číslo: FEI-5382-86165

Vývoj interaktívnej animácie o princípe činnosti mikropočítača

Bakalárska práca

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta elektrotechniky a informatiky

Evidenčné číslo: FEI-5382-86165

Vývoj interaktívnej animácie o princípe činnosti mikropočítača

Bakalárska práca

Študijný program: aplikovaná informatika

Študijný odbor: informatika

Školiace pracovisko: Ústav informatiky a matematiky

Vedúci záverečnej práce: prof. Ing. Ľubica Stuchlíková, PhD.

Bratislava 2020

Dominika Šuhajdová



ZADANIE BAKALÁRSKEJ PRÁCE

Študentka: **Dominika Šuhajdová**
ID študenta: 86165
Študijný program: aplikovaná informatika
Študijný odbor: informatika
Vedúca práce: prof. Ing. Ľubica Stuchlíková, PhD.
Miesto vypracovania: Ústav elektroniky a fotoniky

Názov práce: **Vývoj interaktívnej animácie o princípe činnosti mikropočítača**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

Cieľom práce je návrh a vývoj interaktívnej edukačnej animácie vo vybranom programe na báze objektovo orientovaného programovania. Interaktívna animácia bude zobrazovať princíp činnosti mikropočítača a stane sa súčasťou interaktívnych www kurzov pre študentov FEI STU v Bratislave umiestnených na portáli eLearn central. Vytvorená animácia bude súčasne voľne dostupná a voľne šíriteľná v rámci popularizácie vedy a techniky.

Úlohy:

1. Oboznámte sa s princípmi tvorby interaktívnych animácií vo vybranom programe na báze objektovo orientovaného programovania.
2. Vytvorte interaktívnu animáciu o princípe činnosti mikropočítača vo vybranom programe s využitím objektovo orientovaného programovania.
3. Vypracujte používateľskú dokumentáciu a manuály na aktualizáciu animácie.

Zoznam odbornej literatúry:

1. Hrbáček, J. *Komunikace mikrokontroléru s okolím. 1. díl.* Praha : BEN – technická literatura, 1999. 159 s. ISBN 80-86056-42-2.
2. Stuchlíková, Ľ. – Kósa, A. – Jakuš, J. – Šušoliak, M. – Donoval, D. – Hrbáček, J. Interactive animation as a motivation tool. In *EWME 2014. 10th European Workshop on Microelectronics Education: Proceedings; Tallinn; Estonia; May 14-16, 2014.* Danvers : IEEE, 2014, s. 116–119. ISBN 978-147994016-5.
3. Hrbáček, J. *Moderní učebnice programování jednočipových mikrokontrolérů PIC. 2. díl: Stavíme se na vlastní nohy.* Praha : BEN – technická literatura, 2007. 141 s. ISBN 978-80-7300-137-7.
4. ŽÁRA, Ondřej. *JavaScript: programátorské techniky a webové technologie.* Brno: Computer Press, 2015. ISBN 978-80-251-4573-9.

Riešenie zadania práce od: 23. 09. 2019

Dátum odovzdania práce: 01. 06. 2020

Dominika Šuhajdová
študentka

Dr. rer. nat. Martin Drozda
vedúci pracoviska

prof. Dr. Ing. Miloš Oravec
garant študijného programu

ABSTRAKT

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program :	Aplikovaná informatika
Bakalárska práca:	Vývoj interaktívnej animácie o princípe činnosti mikropočítača
Autor:	Dominika Šuhajdová
Evidenčné číslo:	FEI-5382-86165
Vedúci záverečnej práce:	prof. Ing. Ľubica Stuchlíková, PhD.
Miesto a rok predloženia práce:	Bratislava 2020

Táto bakalárska práca sa zaoberá návrhom a vývojom interaktívnej animácie o princípe činnosti mikropočítačov na báze objektovo orientovaného programovania. Interaktívna animácia zobrazuje princíp činnosti jednotlivých komponentov mikropočítača a stane sa súčasťou interaktívnych www kurzov. Cieľovou skupinou sú všetci záujemcovia o problematiku mikropočítačov. V úvode bakalárskej práce sa zaoberáme problematikou mikropočítačov a technológiami pre tvorbu interaktívnych animácií. Opis riešenia je navrhnutý v druhej kapitole. Hlavným výstupom práce je interaktívna animácia, ktorú vie používateľ plne riadiť. Súčasťou bakalárskej práce je používateľská dokumentácia, ktorá používateľom pomáha rýchlejšie sa zorientovať v animácií. Vytvorená animácia bude voľne dostupná a voľne šíriteľná v rámci popularizácie vedy a techniky.

Kľúčové slová: Interaktívna animácia, mikropočítač, popularizácia vedy a techniky,
objektovo orientované programovanie

ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION
TECHNOLOGY

Curriculum:	Applied Informatics
Title of the bachelor thesis:	Development of interactive animation about the principle of microcomputer operation
Author:	Dominika Šuhajdová
Reference number:	FEI-5382-86165
Supervisor:	prof. Ing. Ľubica Stuchlíková, PhD.
Place and year of submission:	Bratislava 2020

This bachelor thesis deals with the design and development of interactive animation about the principle of operation of microcomputers based on object-oriented programming. The interactive animation shows the principle of operation of the individual components of the microcomputer and becomes part of the interactive web courses. The target group is all those interested in the issue of microcomputers. In the introduction to the bachelor's thesis, we deal with the issue of microcomputers and technologies for creating interactive animations. A description of the solution is proposed in the second chapter. The main output of the work is an interactive animation that the user can fully control. Part of the bachelor thesis is user documentation, which helps users to orient themselves more quickly in animation. The created animation will be freely available and freely distributable within the popularization of science and technology.

Key words: Interactive animation, microcomputer, popularization of science and technology, object-oriented programming

Pod'akovanie

Ďakujem doc. Ing. Ľubici Stuchlíkovej, PhD. za odborné vedenie, pripomienky a cenné rady pri vypracovávaní bakalárskej práce.

Čestné vyhlásenie:

Čestne vyhlasujem, že som bakalársku prácu vypracovala samostatne, pod odborným vedením vedúcej bakalárskej práce a použila som len literatúru, ktorú uvádzam v priloženom zozname.

V Bratislave dňa 31.5.2020

.....

(PODPIS)

Obsah

Úvod	1
1. Analýza	2
1.1 Interaktívna animácia	2
1.2 Technológie tvorby animácie	2
1.2.1 Objektovo orientované programovanie	3
1.2.2 HTML5	4
1.2.3 CSS3	4
1.2.4 Javascript	5
1.2.5 JQuery	6
1.2.6 Adobe Flash	6
1.2.7 Java	7
1.3 Mikropočítače	7
1.3.1 Raspberry Pi4 model B	8
1.4 Záver analýzy	9
2. Postup riešenia	10
2.1 Návrh riešenia	10
2.2 Špecifikácia animácie	11
2.3 Implementácia animácie	12
2.3.1 Implementácia úvodnej stránky	12
2.3.2 Implementácia hlavnej časti animácie	13
2.3.3 Implementácia podstránky FAQ	16
2.4 Testovanie funkčnosti kódu	19
2.4.1 Validita súborov	19
2.4.2 Testovanie funkčnosti	21
2.4.3 Zhrnutie výsledkov testov	23
2.5 Optimalizácia kódu	23
2.6 Používateľská dokumentácia	25
2.6.1 Intro	25
2.6.2 Play	26
2.6.3 FAQ	28
Záver	30

Zoznam obrázkov	31
Zoznam skratiek	32
Zoznam použitej literatúry	33
Príloha	I
Štruktúra elektronického média	I

Úvod

Napriek tomu, že technológie majú rýchly vývoj a sú dostupné pre každého, počet ľudí ktorí sú technicky zdatní je stále veľmi nízky a dopyt po expertoch v oblasti technológií neustále rastie.

Mikropočítače, rovnako ako počítače, sú dnes už bežnou súčasťou našich životov a ich použitie nájdeme skoro v každej oblasti od zdravotníctva, cez školstvo až po vedu a techniku. Mikropočítače dostali svoj názov, vďaka svojej veľkosti, obsahujú len elementárne komponenty ale ich funkcionality sa veľmi nelíši od klasických počítačov. Mikropočítače môžu odradiť viacerých používateľov svojím vzhlľadom. Neobsahujú totiž monitor ani klávesnicu, v základnom balíčku dokonca ani puzdro. Tieto vstupno-výstupné zariadenia si používateľ môže napojiť dodatočne, vyžaduje si to však aspoň základné elektrotechnické zručnosti.

Práve požiadavka potreby základných zručností je jedna z mnohých vecí, kvôli ktorým sa mnohí mladí ľudia boja študovať v oblasti IT technológií a elektroniky. K ďalším patrí náročná matematika, fyzika, programovanie,...

Jedným z dôvodov prečo sme sa rozhodli navrhnúť a vyvinúť interaktívnu edukačnú animáciu o princípe činnosti mikropočítačov je zlepšenie predstavy o fungovaní mikropočítačov, pochopenie základných súvislostí, a tým aj príspevok k zvýšeniu záujmu mladých ľudí o túto oblasť.

Animácia bude dostupná všetkým študentom ako aj záujemcom o túto problematiku, bude spĺňať požiadavky kladené na kvalitu animácie. Bude ľahko ovládateľná, jednoduchá na použitie a pochopenie. Používateľ si bude určovať sám tempo štúdia z animácie.

Cieľom tejto bakalárskej práce navrhnúť a vytvoriť interaktívnu animáciu, ktorá vysvetlí používateľovi, čo je mikropočítač a ako fungujú jeho jednotlivé komponenty. Po prejdení animácie bude vedieť používateľ povedať, kde sa aký komponent na mikropočítači nachádza a na čo slúži.

1. Analýza

Aby sme mohli vytvoriť kvalitnú edukačnú interaktívnu animáciu o princípe činnosti mikropočítačov, musíme sa oboznámiť s animáciou, analýzou technológií pre vývoj interaktívnej animácie, kvalitou spracovania už existujúcich interaktívnych animácií a prehľadu súčasného stavu o problematike mikropočítačov.

1.1 Interaktívna animácia

„Až 80% toho čo sa naučíme vnímame zrakom“ (1). Výskumné dôkazy o vzdelávacej účinnosti animácií sú zmiešané. Správne navrhnuté animácie môžu študentom pomôcť rýchlejšie a ľahšie sa učiť. Sú tiež vynikajúcou pomocou pre učiteľov, pokiaľ ide o vysvetlenie ťažkých predmetov. Problémy s predmetmi môžu vzniknúť v dôsledku zapojenia matematiky alebo fantázie (2). Takéto problémy môžu vzniknúť napríklad pri pokusoch, ktoré nie je možné uskutočniť alebo sú náročné na uskutočnenie. Vzhľadom na túto skutočnosť je interaktívna animácia veľkým prínosom najmä v oblasti vzdelávania.

Interaktívne animácie sú, jednoducho povedané animácie, ktoré umožňujú divákovi zúčastňovať sa aktivít na viacerých úrovniach, než je len na pozeraanie (3).

Vďaka interaktívnej animácii používateľ ľahšie pochopí činnosť jednotlivých častí mikropočítača. Aby interaktívna animácia bola efektívna musí byť jednoducho ovládateľná a sprevádzaná písomným vysvetlením. Používateľ musí mať nad interaktívnou animáciou plnú kontrolu, vie meniť rýchlosť prehrávania, smer prehrávania a zobrazovať popis jednotlivých udalostí. Pridanie interaktivity k video - obsahu prináša hlbšie učenie, ktoré ľudom umožňuje skúmať a objavovať vlastným tempom (4).

1.2 Technológie tvorby animácie

Pri interaktívnej animácii sa jedná o simulovanie určitej situácie, kde je potrebné mať vhodný softvér, ktorý nám animáciu spustí. To môže byť v niektorých prípadoch nevýhodou. Interaktívnu animáciu môžeme vytvoriť viacerými spôsobmi ako napríklad pomocou Java, Flash, JavaScript, HTML, CSS. Na základe bližšej analýzy týchto spôsobov sme sa rozhodli využiť kombináciu HTML, CSS, Javascript a JavaScriptovú knižnicu JQuery.

Existujú aj iné alternatívne možnosti. Jednou z nich je Flash animácia, ktorá využíva skriptovací jazyk Actionscript (podobný Javascript-u). Napriek tomu, že sa jednoducho

ovláda a pracuje priamo s vektorovou grafikou je v našej animácii nevýhodná. Dôvodov je hneď niekoľko: táto technológia nie je podporovaná viacerými prehliadačmi, nehodí sa na prezentáciu dlhých textov a je náročnejšia na výkon počítača.

1.2.1 Objektovo orientované programovanie

„Objektovo orientované programovanie alebo objektové programovanie (skratka OOP, z angl. „Object - Oriented Programming“) je programovacia paradigma založená na používaní dátových štruktúr nazývaných objekty a ich interakcie na vývoj aplikácií. Princípy objektového programovania boli rozpracované už v 70. rokoch 20. storočia, no širšie sa vo vývoji softvéru začalo uplatňovať až koncom 20. storočia. V súčasnosti existuje veľa jazykov využívajúcich princípy OOP“ (5).

Každý objekt má svoje vlastnosti ako napríklad šírka a výška, metódy a funkcie, ktoré určujú ako sa daný objekt správa (6).

„Z obsahového hľadiska vlastnosti typu trieda sú vo svojej podstate položky typu záznam. Metódy a udalosti sú svojím charakterom funkcie a procedúry. Udalosťou sa nazýva každá zmena stavových veličín, napr. Click, DoubleClick, stlačenie klávesy na klávesnici, impulz z časovača, Zmena veľkosti okna, zatvorenie okna. Základom objektového programovania je dátový typ trieda. Dátový typ trieda je odvodený dátový typ a vychádza z dátového typu štruktúra“ (7).

Objektovo orientované programovanie má viacero výhod. OOP pomáha k zlepšeniu prehľadnosti kódu, jeho znova použitiu a jednoduchšej aktualizácii (8). Vďaka objektom nepotrebujeme meniť celý kód ale stačí aktualizovať len daný objekt. Tým, že kód je rozdelený do viacerých tried je aj ľahšie čitateľný a zrozumiteľnejší. Ďalšou skvelou výhodou je flexibilita kódu, vďaka čomu môžeme jeden kód použiť vo viacerých projektoch bez potreby úpravy kódu alebo len z minimálnymi úpravami. Vďaka svojej štruktúre nám dokáže ušetriť veľa času s programovaním zložitejších programov. To však môže byť nevýhodou v jednoduchších programoch, kde nám vytvorenie triedy objektu zaberie viac času ako priamy zápis objektu. Pri rozhodovaní či je vhodné použiť objektovo orientované programovanie, je potrebné brať do úvahy koľko objektov v danom projekte bude a ako často bude aktualizovaný.

1.2.2 HTML5

„HTML (HyperText Markup Language) je značkovací jazyk určený na vytváranie webových stránok a iných informácií zobraziteľných vo webovom prehliadači“ (9).

HTML5 je novšia verzia jazyka HTML. Je podporovaný všetkými prehliadačmi a práca s ním je jednoduchá. Nie je potrebná žiadna inštalácia. Čítanie a upravovanie je možné priamo v textovom editore.

HTML tvorí základnú štruktúru stránky a hovorí prehliadaču ako budú jednotlivé prvky usporiadané. Obsahuje tzv. tagy webstránky, ktoré určujú základný vzhľad textu, nadpisov atď. ale môžu obsahovať aj základné stavebné prvky ako napríklad odsek, odkaz a pod. Tieto tagy používateľom nie sú zobrazené vďaka softvéru prehliadača, ktorý tento kód prekladá do čitateľnej podoby (10). „Tento jazyk bol vytvorený v roku 1991 tvorcom Timom Berners-Lee a vydaný v roku 1995“ (11).

Všetky programovacie jazyky (napr. Java, C, PHP, Javascript...) sa vyznačujú tým, že dokážu vytvárať podmienky ako napr. ak sa stane x urob y. Toto HTML nedokáže a preto nie je programovacím jazykom, napriek tomu ho ľudia mylne ako programovací jazyk označujú. HTML iba zobrazuje určité elementy (texty, obrázky, videá, tabuľky...) (12).

Jednou z nevýhod je, že pokiaľ zabudneme ukončiť tag alebo máme chybu v syntaxe, tak nás na to editor neupozorní. Čím viac podstránok, tým menšia šanca si takúto chybu všimnúť. Rovnako sa nám môže stať, že nám budú chýbať určité prepojenia medzi podstránkami. Ďalšou nevýhodou je, že validitu stránky musíme otestovať osobitne. Textový editor to za nás nespraví. Tým, že HTML nie je programovací jazyk, môžeme ním prvky iba zobrazovať ale nemôžeme overiť funkcionality vyhodnocovania testov. K tomu, aby sme mohli overiť funkcionality, musíme k HTML pridať aj programátorský jazyk (ideálne JavaScript alebo PHP). Veľkou výhodou je množstvo kurzov, návodov a tutoriálov ako sa ho naučiť a ako s ním správne pracovať. Jeden z najlepších návodov je na stránke www.w3school.com.

1.2.3 CSS3

CSS je skratka pre Cascading Style Sheet, inak nazývané aj ako kaskádové štýly. Je to rozšírenie HTML (13). Určuje dizajn a štylizáciu jednotlivých prvkov HTML kódu (farba, zaoblenie, posun, počet stĺpcov...).

Verzia CSS3 je rozšírenou verziou CSS2 doplnenou o animácie a efekty ako napríklad tieň písma alebo transformácia. CSS3 a Javascript majú niektoré funkcie podobné ako napr. už spomínaná transformácia. Rozdiel je, že CSS analyzuje a vykresľuje rýchlejšie. Chyba JavaScriptu môže zastaviť všetky JavaScriptové súbory na stránke. Nesprávna vlastnosť CSS alebo vynechanie bodkočiarky iba preskočí vlastnosť a pokračuje ďalej (14). Pri písaní CSS kódu „rovnať ako pri HTML, nie je potrebná žiadna inštalácia a pracuje sa z textového editora. Obrovskou výhodou je ovládanie viacerých prvkov HTML dokumentu pomocou jedného CSS príkazu. CSS využíva dedičnosť (15). Vďaka dedičnosti nie je potrebné, každému elementu nastavovať rovnaké vlastnosti, napr. farba písma.

Ďalšou výhodou, ktorá je však aj jeho nevýhodou je jednoduchosť. CSS je jednoduché sa naučiť, avšak je potrebné striktné dodržiavanie disciplíny písania kódu. Tým máme na mysli štruktúru zapisovania a označovanie prvkov (16). V opačnom prípade sa vieme v kóde veľmi jednoducho stratiť a jedna deklarácia nám môže prepisovať inú, čo je tiež jednou z nevýhod. CSS vyžaduje aj podporu prehliadačov, čo už v súčasných prípadoch nie je žiadny problém.

1.2.4 Javascript

Vzhľadom k tomu, že naša bakalárska práca má byť naprogramovaná v objektovo orientovanom jazyku je Javascript skvelou voľbou. „Javascript je objektovo orientovaný skriptovací programovací jazyk, používaný pri tvorbe dynamických prvkov stránky. Pôvodne ho vyvíjal Brendan Eich zo spoločnosti Netscape Communications pod názvom Mocha, neskôr pod menom LiveScript, ale vďaka vtedajšej popularite Javy bol premenovaný na Javascript. Neexistuje žiadny vzťah medzi jazykmi Java a JavaScript; ich najväčšia podobnosť je v syntaxi založenej na spoločnom predkovi - jazyku C“ (17).

Javascript sa spúšťa na strane klienta a na rozdiel od HTML a CSS nám tento programovací jazyk ohlásí akúkoľvek chybu v kóde. Výhodou JavaScriptu sú menšie nároky na výkon servera, na ktorom sú stránky umiestnené (18). Nevýhodou JavaScriptu je že používateľ môže JavaScript vo svojom prehliadači zakázať. Ďalšou dôležitou nevýhodou je ako sme už spomínali vyššie, chyba JavaScriptu môže zastaviť všetky JS na stránke.

V posledných rokoch vzniklo mnoho rozšírení JavaScriptu, z ktorých sa niektoré stali samostatnými vývojárskymi nástrojmi a dnes už na webových stránkach samotný JavaScript nájdeme len výnimočne (19). Takýmito rozšíreniami sú napríklad: Angular, React, Vue.js, Node.js a ďalšie.

1.2.5 JQuery

„Jquery je cross-browser JavaScript knižnica, ktorá kladie dôraz na interakciu medzi JavaScriptom a HTML. Vydal ju v januári 2006 v BarCamp NYC vývojár John Resig. JQuery je slobodný a otvorený software pod MIT licenciou“ (20). Používa sa na viac ako 34% webstránok a aktuálne je to najpoužívanější knižnica. Jeho syntax je navrhnutá tak aby nám umožnila vytváranie animácií, jednoduchšiu navigáciu dokumentu, spracovanie udalostí, vývoj Ajax aplikácií, výber DOM elementov (21). Vývojári majú možnosť na základe tejto Javascript knižnice vytvárať množstvo pluginov, je rýchla a presná (22).

Nedá sa jednoznačne definovať či je vhodnejšie používať Javascript alebo JQuery. Obidve možnosti ponúkajú vytvorenie rovnakých efektov. JQuery je však jednoduchšie a častokrát vďaka nemu vieme s menším počtom riadkov vytvoriť úplne rovnaký efekt ako pri JavaScripte (23).

Jquery disponuje množstvom výhod ako je napr. jednoduché používanie, kvalitne spracovaná dokumentácia, množstvo voľne dostupných tutoriálov, ktoré môžete nájsť aj na YouTube, prípadne si viete zaplatiť kurz na stránkach ako sú www.learn2code.sk alebo www.it-academy.sk. Ďalšou jeho skvelou vlastnosťou, oproti iným frameworkom je, že sa striktne drží vo svojom vlastnom priestore a nemodifikuje nič, čo mu nepatrí (24).

1.2.6 Adobe Flash

Adobe Flash je staršia multimediálna softvérová platforma, pomocou ktorej môžeme vytvoriť animácie, webové aplikácie, mobilné aplikácie, hry a videoprehrávače s internetovým prehliadačom (25).

Jednou z výhod programu Adobe Flash je využívanie vektorovej grafiky, ktorá nepodlieha deformáciám pri manipulácii s veľkosťou obrázkov (26). Ďalšou výhodou je, že náš výsledný kód vieme uzamknúť, tak aby ho nikto nemohol zneužiť. Ovládanie Adobe Flash je jednoduché, reaguje na činnosť používateľa (pohyb kurzora myši, stlačenie klávesnice a pod.). Zvuky je možné ukladať v MP3. Je omnoho prepracovanejší ako HTML (27). Na vývoj webových animácií v program Flash sa využíva objektovo orientovaný programovací skriptovací jazyk ActionScript, ktorý je veľmi podobný JavaScriptu (28).

Oproti HTML má Flash mnoho nevýhod. Flash má vysoké nároky na výkon počítača (29). Toto sa môže prejavovať pomalým načítavaním animácie alebo trhaným zobrazovaním. Dá sa to čiastočne eliminovať ale zaplatíme za to znížením kvality animácie.

Pri práci s textami je veľmi nevhodný, pretože jednotlivé znaky textu interpretuje ako vektorové objekty a pri každom posune prepočítava veľké množstvo parametrov (30). Je náročný na aktualizáciu. Pri každej aktualizácii musíme vo väčšine prípadov vygenerovať celú scénu na novo. V tomto je jednoduchší JavaScript, kde nám stačí zmena konkrétneho objektu. Pri probléme s Flash animáciou nám málokedy pomôže obnova webstránky, pretože proxy serveri si často pamätajú pôvodné scény. Najväčšou a rozhodujúcou nevýhodou Flash je ukončenie jeho podpory v decembri 2020 (31).

1.2.7 Java

Java je univerzálny, objektovo orientovaný a najstabilnejší programovací jazyk (32). Využíva sa pri tvorbe aplikácií, hier, interaktívnych animácií a pri pripojeniach k databázam.

Java má množstvo výhod. Pracuje na rôznych platformách. Je to jeden z najpopulárnejších programovacích jazykov na svete. Je bezplatná, rýchla, bezpečná a pretože uľahčuje programátorom prechod z C++ a C# na Java a opačne, má obrovskú podporu komunity (33).

Nevýhodou Javy je výrazne pomalšia rýchlosť aplikácie a nekompatibilita so staršími verziami Javy.

1.3 Mikropočítače

„Mikropočítač je označenie pre počítač s mikroprocesorom (teda s CPU v jednom integrovanom obvode). Mikropočítač vznikne tak, že sa k mikroprocesoru (CPU) pridajú aspoň pamäťové a vstupno-výstupné obvody, prípadne periférne zariadenia, aby bol mikropočítač schopný samostatnej práce“ (34).

Mikropočítače sa vyznačujú tým, že sú omnoho menšie a ich operačná pamäť nie je extrémne veľká oproti klasickým počítačom. V niektorých aplikáciách môžu byť pomalšie než štandardné počítače, napriek tomu dokážu všetko čo dokáže aj klasický počítač (35). Obrovskou výhodou je cenová dostupnosť mikropočítačov.

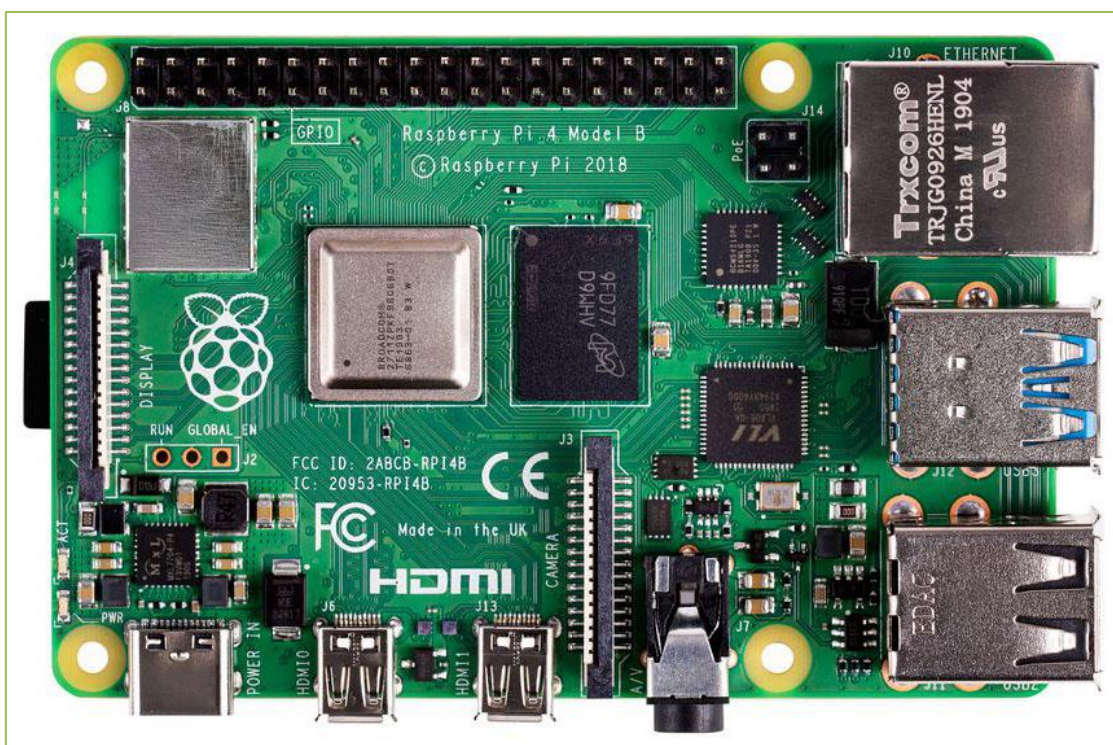
Dnešné mikropočítače využívame takmer v každej oblasti. Zjednodušujú nám život, pomáhajú nám, učia nás. Veľmi obľúbené sú v robotike, v hračkách, v priemysle ale aj v lekárstve. V našej animácii použijeme ako model princípu fungovania mikropočítač Raspberry Pi, ktorý bol pôvodne navrhnutý ako pomôcka pri výučbe a študenti sa mali na

ňom učiť programovať, ale už pri výrobe bolo jasné, že tento malý mikropočítač zvládne omnoho viac (36).

Pre prácu s mikropočítačmi na internete už nájdeme množstvo materiálov, kníh, manuálov, tutoriálov na YouTube ale žiadnu interaktívnu animáciu, ktorá by opisovala princíp činnosti mikropočítačov.

1.3.1 Raspberry Pi4 model B

Raspberry Pi 4 model B, ktorý používame ako ukážku v našej animácii (Obr. 1), je najnovší model mikropočítača od spoločnosti Raspberry Pi. Raspberry Pi 4 Model B bol uvedený na trh v júni 2019 s procesorom ARM Cortex-A72 s frekvenciou 1,5 GHz, zabudovaným rozhraním 802.11ac Wi-Fi, Bluetooth 5, plným gigabitovým Ethernetom (priepustnosť nie je obmedzená), dvoma portami USB 2.0, dvoma portami USB 3.0 a podporou duálneho monitora prostredníctvom párov portov micro HDMI (HDMI Type D) až do rozlíšenia 4kB. Zariadenie Pi 4 je napájané aj cez port USB-C, ktorý umožňuje dodávať ďalší prúd periférnym zariadeniam, ak sa používa s príslušnou jednotkou PSU (37).



svojom prvom miniatúrnom, lacnom a jednoduchom počítači. Jeho cieľom bolo vytvoriť malý počítač, finančne dostupný pre všetkých študentov, aby sa mohli učiť programovať a pritom sa rodičia nebáli, že dieťa mikropočítač zničí. Pôvodne chcel vytvoriť iba zopár kusov, ktoré by študentom rozdali a podľa toho, čo zaujímavé študent s mikropočítačom spraví, mali byť títo študenti prijatí do školy, na ktorej vyučoval. Tento mikropočítač už od svojich začiatkov prináša viac možností než len ako vzdelávacia pomôcka (38).

Amatérski technickí nadšenci používajú dosky Raspberry Pi ako mediálne centrá, súborové servery, retro herné konzoly, a iné. Je to však len málo z toho, čo je Raspberry Pi dokáže. Existujú stovky projektov, kde ľudia používali Raspberry Pi mikropočítač na zostavenie robotov, inteligentných zrkadiel, na fotografovanie vesmíru, na vykonávanie experimentov na Medzinárodnej vesmírnej stanici a pod.

Keďže je Raspberry Pi 4 rýchlejší, schopný dekódovať 4K video, využívať rýchlejší ukladací priestor cez USB 3.0 a rýchlejšie sieťové pripojenie prostredníctvom skutočného gigabitového Ethernetu, má mnoho nových využití. Je to tiež prvý Raspberry Pi, ktorý podporuje dve obrazovky naraz - až duálne 4K @ 30 obrazoviek -to je výhoda pre ľudí, ktorí chcú viac miesta na pracovnej ploche (39).

1.4 Záver analýzy

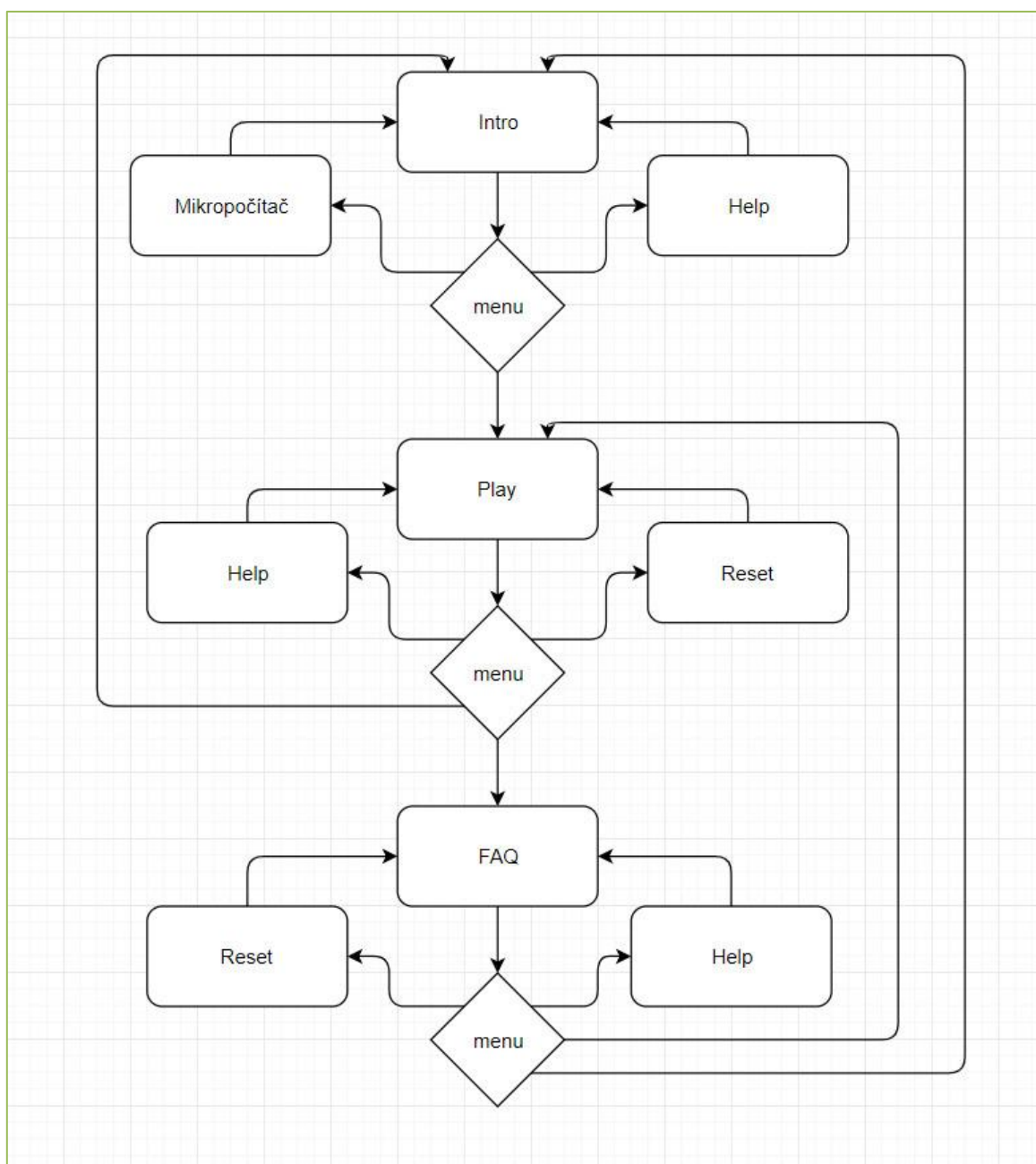
Na základe uvedenej analýzy sme sa rozhodli kvôli nedostupnosti postačujúcej vizualizácií princípu činnosti mikropočítačov vytvoriť interaktívnu animáciu pomocou technológií HTML5, CSS3, JQuery a JavaScript. Dôvod tohto výberu je jednoduchý. Táto technológia je dostupná všetkým používateľom bez potreby akejkol'vek ďalšej inštalácie, či sťahovania súborov. Interaktívna animácia je najlepším spôsobom výučby, hlavne kvôli jednoduchému ovládaniu a lepšiemu zapamätaniu si učiva a mikroprocesory majú široké možnosti využitia.

2. Postup riešenia

Na základe vyššie uvedenej analýzy sme sa rozhodli pre nasledovný návrh riešenia.

2.1 Návrh riešenia

Interaktívna animácia o princípe činnosti mikropočítačov bude vytvorená na báze objektovo orientovaného programovania. Nasledujúci UML diagram na Obr. 2 zobrazuje postupnosť krokov, ktoré budeme môcť v aplikácii vykonať.



Obr. 2 UML diagram aplikácie

Cieľom tejto animácie je dostatočne opísať činnosti jednotlivých komponentov mikropočítača a ich fungovania ako celku. Používateľ si bude môcť sám určiť poradie v akom bude animácia prebiehať. Animácia bude vypracovaná v 2 oknách + menu, pričom bude možné zväčšovať text v animácií podľa vlastnej potreby. Výsledná animácia bude pre funkčnosť vo všetkých moderných prehliadačoch naprogramovaná v jazyku Javascript, JQuery, HTML5 a CSS3.

2.2 Špecifikácia animácie

V časti „intro“ si používateľ bude môcť pozrieť základné informácie o mikropočítačoch a pomocou tlačidiel sa bude môcť rozhodnúť ako bude animácia pokračovať. Pomocou tlačidla ovládanie si používateľ bude môcť pozrieť celý používateľský manuál a ovládanie aplikácie.

V sekcii kontrolný test si používateľ bude môcť otestovať svoje vedomosti bez ohľadu na množstvo preštudovaných komponentov. Používateľ si bude vedieť bez problémov otestovať svoje vedomosti pred prácou s animáciou a porovnať si ich so svojimi vedomosťami po bližšom preskúmaní animácie. Vďaka tejto funkcii budeme vedieť kontrolovať či aplikácia splňa cieľ.

V hlavnej aplikácii budeme vidieť horizontálne menu a pod ním dve základné okná: okno, v ktorom bude zobrazený mikropočítač Raspberry Pi4 s tlačidlami, ktoré sa zobrazia až keď používateľ prejde myšou na toto tlačidlo a okno so základnými informáciami o mikropočítači Raspberry Pi4. Po zakliknutí komponentu sa časť zobrazujúca mikropočítač zmenší a časť informácie sa rozšíri. V ľavej časti zobrazujúcej mikropočítač ostane zakliknuté tlačidlo vysvietené na zeleno, čím používateľ získa informáciu kde sa daný komponent na mikropočítači nachádza. V pravej časti zobrazujúcej informácie sa obsah bude meniť podľa vybraného komponentu.

V podstránkach „FAQ“ a „Play“ si používateľ bude vedieť prispôbovať veľkosť písma.

2.3 Implementácia animácie

Na základe návrhu riešenia sme sa rozhodli vytvoriť 3 podstránky. Ako sme už videli v návrhu riešenia, každá podstránka je špecifická a má iné rozloženie, preto sme pre každú podstránku museli vytvoriť vlastné HTML.

2.3.1 Implementácia úvodnej stránky

Úvodná stránka animácie je veľmi jednoduchá. Obsahuje nadpis, v ktorom slovo „Mikropočítač“ funguje ako tlačidlo, ktoré po kliknutí zobrazí dialógové okno s vysvetlením základného princípu mikropočítačov. Pod nadpisom sa nachádza menu. Dve z troch tlačidiel v menu fungujú ako odkaz na ostatné podstránky. Tlačidlo „Help“ nám rovnako ako tlačidlo „Mikropočítač“ otvorí dialógové okno s informáciami ako animáciu ovládať. Aby úvodná stránka dobre vyzerala ako pozadie stránky sme vložili obrázok mikropočítača, ktorý je voľne použiteľný bez potreby uvedenia zdroja.

Napriek tomu, že na tejto podstránke máme iba jeden typ dynamického prvku (dialógové okno) rozhodli sme sa ho vložiť do externého .js, pretože je rovnaký pre všetky podstránky. Aby sme nemuseli vytvárať kompletne nový kód na tvorbu dialógového okna použili sme voľne dostupné knižnice z webovej stránky <https://code.jquery.com/ui/>, ktoré sme vložili do HTML hlavičky ako odkaz na stránku (Obr. 3) a následne sme spravili len jednoduchú úpravu (Obr. 4), ktorou sme programu povedali, kedy má otvoriť dialógové okno.

Výsledný dizajn a zobrazenie webstránky je možné vidieť na obrázkoch v sekcii používateľská príručka.

```
3 <head>
4   <meta charset="utf-8">
5   <meta name="author" content="Dominika Suhajdova">
6   <title>Raspberry Pi4 - interaktívna animácia</title>
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <link Rel="stylesheet" type="text/css" href="css/style_index.css">
9
10  <script src="https://code.jquery.com/jquery-1.10.2.js"></script>
11  <script src="https://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
12  <link rel="stylesheet" href="https://code.jquery.com/ui/1.10.4/themes/smoothness/
    jquery-ui.css">
13 </head>
```

Obr. 3 Kód HTML hlavičky


```

1  /*-----help-----*/
2  $(document).ready(function(){
3      $( "#control" ).dialog({ autoOpen: false });
4      $( "#govern" ).click(function() {
5          $( "#control" ).dialog( "open" );
6      });
7
8      $( "#introduction" ).dialog({ autoOpen: false });
9      $( "#microPC" ).click(function() {
10         $( "#introduction" ).dialog( "open" );
11     });
12 });|

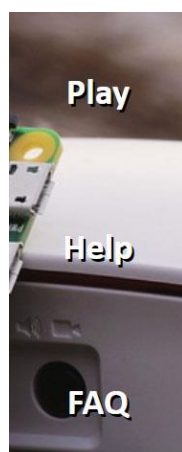
```

Obr. 4 JS kód pre otvorenie dialógového okna

2.3.2 Implementácia hlavnej časti animácie

Do hlavnej časti animácie sa dostane používateľ prostredníctvom menu v úvodnej stránke (Obr. 5). Hlavná časť animácie je zložitejšia a obsahuje viacero dynamických prvkov. Táto podstránka je rozdelená na tri časti.

V hornej časti je zobrazené horizontálne menu. Oproti menu v úvodnej stránke sa líši svojím rozložením a dizajnom. Menu na úvodnej podstránke je zobrazené na Obr. 5 a menu v hlavnej časti animácie je zobrazené na Obr. 6. HTML kód menu je rovnaký ako v úvode, zmenil sa iba CSS kód. Taktiež sú tam pridané 2 tlačidlá: „Reset“ a „0/16“.



Obr. 5 Menu v úvodnej stránke



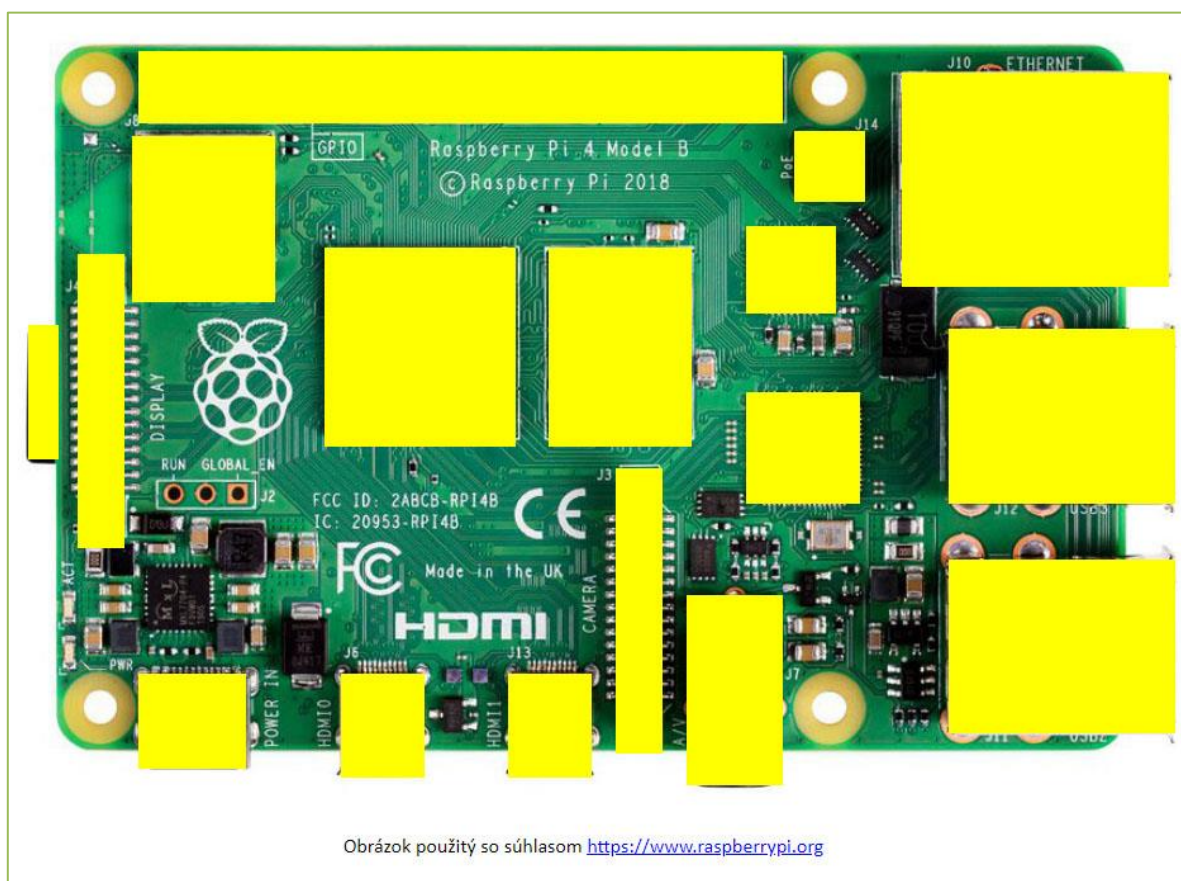
Obr. 6 Menu v hlavnej časti animácie

Tlačidlo „0/16“ neplní žiadnu funkciu, iba zobrazuje počet preštudovaných komponentov. Bolo však pridané ako tlačidlo, kvôli rovnomernému rozloženiu menu. Aby to

používateľa nemiatlo, pomocou CSS kódu sme na toto konkrétne tlačidlo nastavili kurzor myši „text“. Keď používateľ prejde kurzorom myši na toto tlačidlo kurzor sa zmení z cursor: pointer na cursor: auto. Tlačidlo „Reset“ obnoví túto webstránku. Počet preštudovaných komponentov sa nastaví na 0.

Pod menu vľavo máme sekciu zobrazujúcu obrázok mikropočítača Raspberry Pi 4 model B. Pod týmto obrázkom je vyjadrenie súhlasu použitia tohto obrázku. Na obrázku sú vytvorené sekcie predstavujúce jednotlivé komponenty. Tieto sekcie majú nastavenú nepriehľadnosť na 0, tzn. nie je ich vidno. Zobrazia sa až po prejdení myšou na tento komponent.

Postup vytvárania animácie bol nasledovný. Ako prvé sme si vytvorili jednotlivé komponenty a umiestnili sme ich na obrázok zobrazujúci mikropočítač Raspberry Pi4 (Obr. 7).



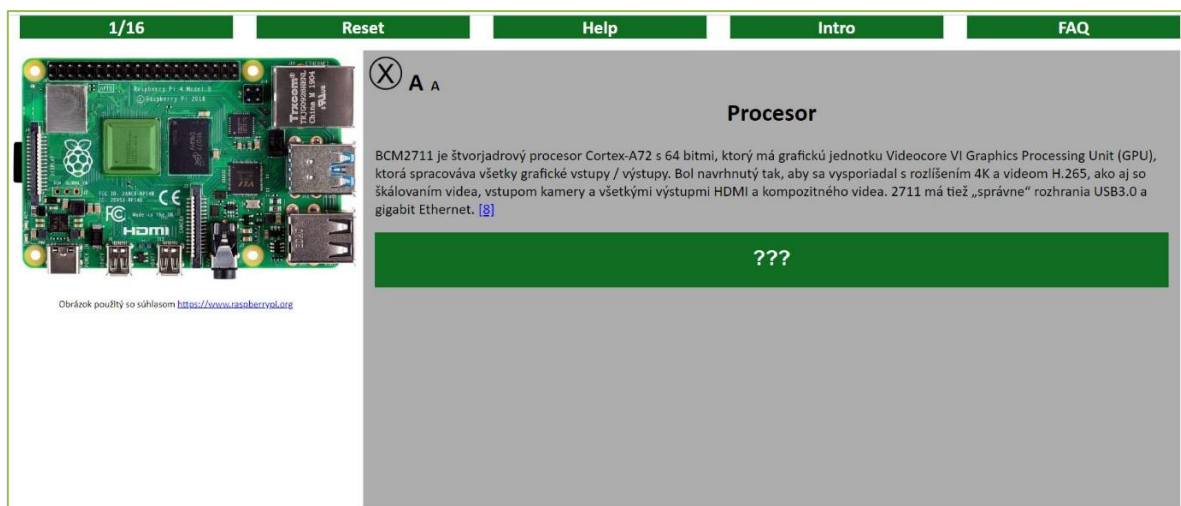
Obr. 7 Ľavá časť animácie - zobrazenie komponentov bez nastavenia priehľadnosti

Aby to lepšie vyzeralo a používateľ mal v komponentoch prehľad, sme týmto sekciám pridali priehľadnosť s tým, že tento komponent sa zobrazil iba keď naň používateľ prišiel

Následne sme vytvorili pravú časť, kde sa zobrazujú informácie. Ako úvodné informácie sme zvolili práve informácie o tom, čo je Raspberry Pi a jeho špecifikáciu. V HTML sme si vytvorili jednotlivé sekcie, ktoré sa vďaka nastaveniam v JavaScripte a CSS zobrazujú až po kliknutí na určitý komponent.

15

Po kliknutí na ľubovoľný komponent sa ľavá strana animácie zmenší a pravá strana zobrazujúca informácie sa zväčší, tak aby čítanie bolo príjemnejšie (Obr. 9). Na ľavej strane ostane zelenou farbou vyznačený komponent, ktorý práve študujeme a na pravej strane sa informácie zmenia na konkrétne informácie o danom komponente. Počet preštudovaných komponentov sa potom zvýši.



Obr. 9 Zmena animácie po kliknutí na komponent

V pravej časti si vieme zmenšovať a zväčšovať písmo a tlačidlom „X“ vrátiť animáciu do pôvodnej veľkosti a základných informácií.

Články a obrázky čerpané z rôznych stránok majú pripojený odkaz na zdroj. Zdroj sa otvára v novom okne, aby nám to nerušilo animáciu.

Všetky dynamické prvky ako napr. zväčšovanie alebo zmenšovanie jednotlivých okien, zobrazovanie rozširujúcich informácií, počítanie preštudovaných komponentov, zmena veľkosti písma a pod. sú vytvorené pomocou CSS a JavaScript.

2.3.3 Implementácia podstránky FAQ

Podstránka „FAQ“ má odlišnú štruktúru ako predchádzajúce dve podstránky. Menu je vertikálne umiestnené v ľavej hornej časti. Obsahuje rovnaké tlačidlá ako v hlavnej časti animácie s výnimkou tlačidla „0/16“. Toto tlačidlo je nám tu nepotrebné.

Vpravo od menu máme zobrazený test. V ľavej hornej časti testu sa nám spolu s načítaním stránky spúšťa časovač. Tento časovač sa zastaví až po odoslaní testu. V strednej hornej časti sa po odoslaní testu zobrazuje počet správnych odpovedí z celkového počtu otázok a v pravej hornej časti máme tlačidlá na zmenu veľkosti písma.

Test sa generuje pomocou JavaScriptu. Tu sme si vytvorili premennú obsahujúcu otázky – „question“, možnosti - „answers“ a správnu odpoveď – „correctAnswer“ (Obr. 10).

```
30 //test otázky
31 var myQuestions = [
32 {
33   question: "Aký operačný systém používa Raspberry Pi4?",
34   answers: {
35     a: 'Linux',
36     b: 'Raspbian',
37     c: 'Windows',
38     d: 'Android'
39   },
40   correctAnswer: 'b'
41 },
42 {
43   question: "Aké sú základné druhy Ethernet?",
44   answers: {
45     a: 'Cat5e a Cat6a',
46     b: 'hrubý ethernet 10base5, tenký ethernet 10Base2, UTP 10baseT',
47     c: 'Fast Ethernet, Gigabit Ethernet'
48   },
49   correctAnswer: 'b'
50 },
51 {
52   question: "Akú prenosovú rýchlosť má USB 2.0? ",
53   answers: {
54     a: '1,5 Mb/s',
55     b: '12 Mb/s',
56     c: '480 Mb/s',
57     d: '5 Gb/s'
58   },
59   correctAnswer: 'c'
60 },
61 {
62   question: "Akú prenosovú rýchlosť má USB 3.0?",
63   answers: {
64     a: '5 Gb/s',
65     b: '10 Mb/s',
66     c: '12 Mb/s',
67     d: '480 Mb/s'
68   },
69   correctAnswer: 'a'
70 },
71 {
72   question: "Ako odlíšiť USB 2.0 port od USB 3.0 portu?",
73   answers: {
74     a: 'USB 3.0 port má modrý konektor, USB 2.0 má čierny konektor',
75     b: 'USB 3.0 má iný tvar',
76     c: 'USB 3.0 port má čierny konektor, USB 2.0 má modrý konektor',
77     d: 'USB 2.0 má iný tvar'
78   },
79   correctAnswer: 'a'
80 },
81 ]
```

Obr. 10 Premenná obsahujúca otázky, odpovede a správnu odpoveď

Následne sme si pomocou funkcie „showQuestions“ (Obr. 11) zobrazili test vo webovom prehliadači.

```

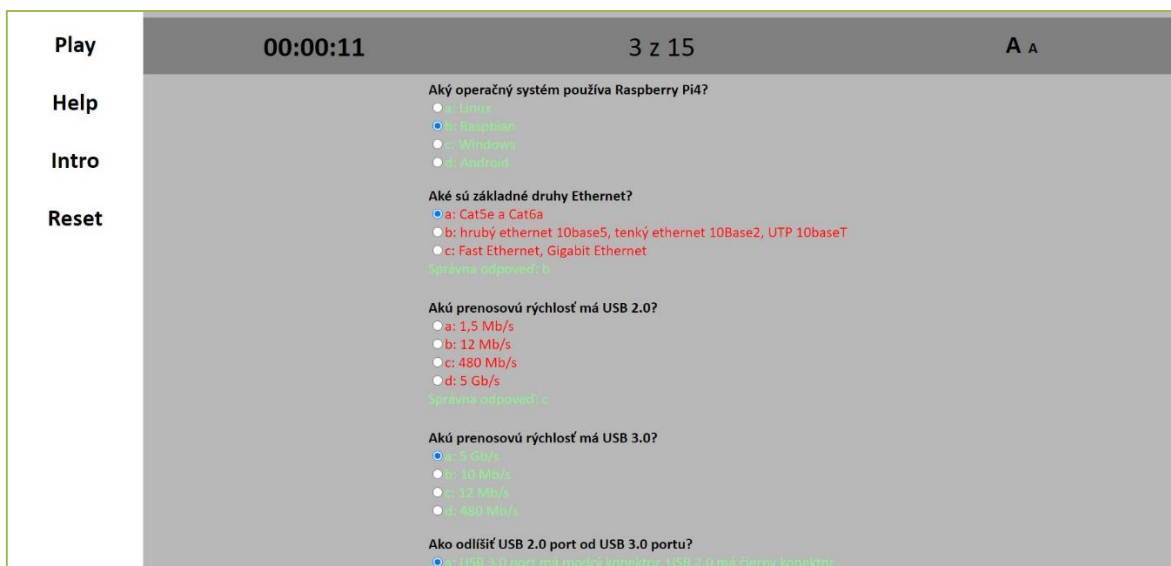
181 var quizContainer = document.getElementById('quiz');
182 var resultsContainer = document.getElementById('results');
183 var submitButton = document.getElementById('submit');
184
185 generateQuiz(myQuestions, quizContainer, resultsContainer, submitButton);
186
187 //generovanie testu
188 function generateQuiz(questions, quizContainer, resultsContainer, submitButton){
189
190     //zobrazenie testu
191     function showQuestions(questions, quizContainer){
192         // vypis odpovede
193         var output = [];
194         var answers;
195
196         //celý test
197         for(var i=0; i<questions.length; i++){
198
199             // list odpovedi
200             answers = [];
201
202             // zobraz celý test
203             for(letter in questions[i].answers){
204
205                 // ...pridaj butony
206                 answers.push(
207                     '<label>'
208                     + '<input type="radio" name="question'+i+'" value="'+letter+'"'>'
209                     + letter + ': '
210                     + questions[i].answers[letter]
211                     + '</label><br>'
212                 );
213             }
214
215             // pridaj otázky a odpovede
216             output.push(
217                 '<div class="question">' + questions[i].question + '</div>'
218                 + '<div class="answers">' + answers.join('') + '</div>'
219                 + '<div class="correctAns">Správna odpoveď: ' + questions[i].correctAnswer + '</div><br>'
220             );
221         }
222
223         //document.getElementById(correctAns).style.display = 'none';
224
225         // finally combine our output list into one string of html and put it on the page
226         quizContainer.innerHTML = output.join('');
227     }
228 }
229

```

Obr. 11 Funkcia zobrazujúca test

Po odoslaní testu, pomocou funkcie „showResults“ skontrolujeme, či označené odpovede sú rovnaké ako správne odpovede a podľa toho vyhodnotíme otázku ako správnu alebo nesprávnu. Pri nesprávnych odpovediach sa otázka vyfarbí na červeno a zobrazí sa k nej zeleným písmom správna odpoveď (Obr. 12). Správna odpoveď vyfarbí otázku na zeleno. Ak používateľ neodpovedal vyhodnotí ju ako zlú odpoveď.

Výhodou tohto kódu je, že ak chceme otázky pridať, odobrať alebo zmeniť postačí nám zmeniť premennú „myQuestions“.



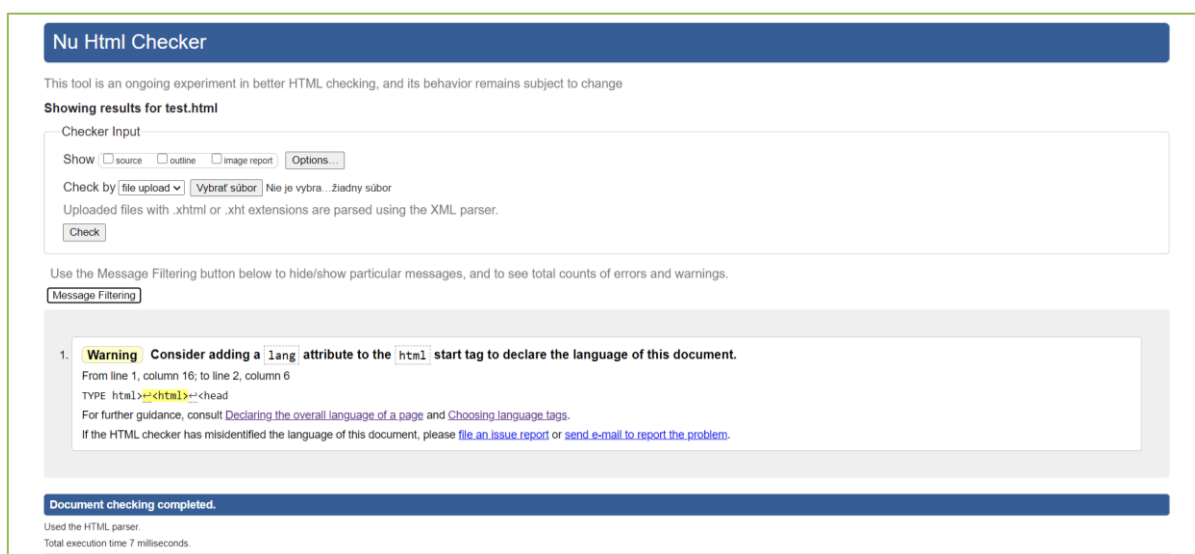
Obr. 12 Vyhodnotenie testu

2.4 Testovanie funkčnosti kódu

HTML a CSS nás neupozorňujú na chyby pri písaní kódu. O to dôležitejšie je spraviť samotné testovanie. Pri testovaní sa zameriavame na funkčnosť animácie v rôznych prehliadačoch a v rôznych zariadeniach.

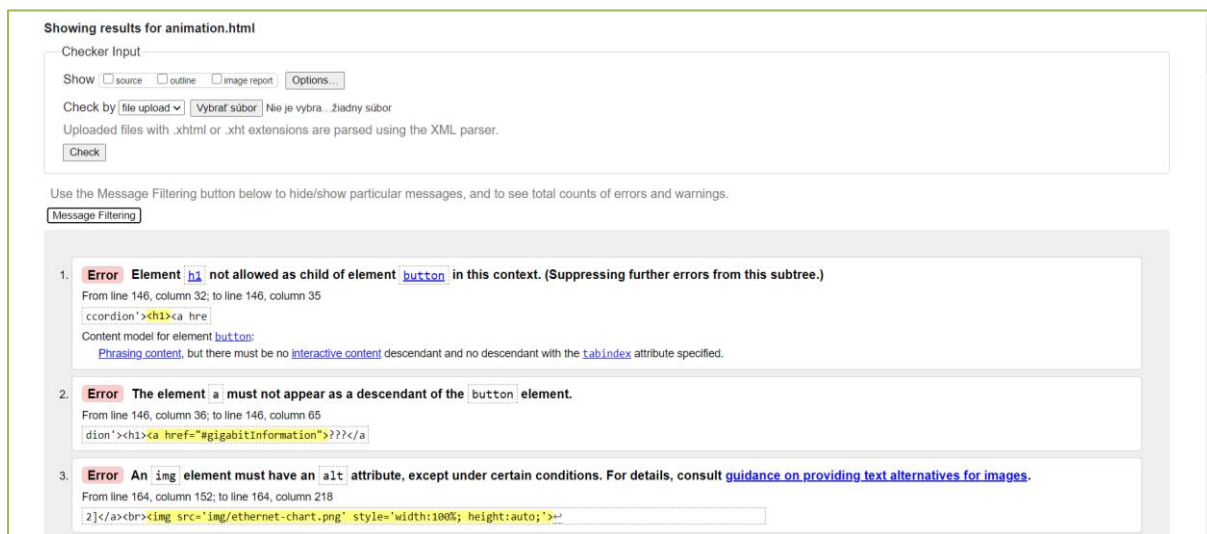
2.4.1 Validita súborov

V prvej fáze testovania sme skontrolovali validitu všetkých HTML a CSS súborov. Táto fáza v podstránkach „intro.html“ a „FAQ“ prebehla úspešne. Jediné čo bolo treba pridať bol jazyk webstránky (Obr. 13), ale toto upozornenie nijak neovplyvňovalo funkčnosť našich stránok.



Obr. 13 Kontrola validity HTML

V podstránke „animation.html“ bola pravdepodobnosť chýb väčšia z dôvodu dlhšieho kódu. Validita tejto podstránky nám zobrazila chýb viac (Obr. 14).



Obr. 14 Validita podstránka "animation.html"

Najpodstatnejšou chybou bola chyba duplicitného id. Toto sme vyriešili novým názvom id. Ďalšou chybou bolo, že prvok <a> nesmie byť potomok prvku <button>. V tomto prípade nám <a> slúžilo ako odkaz na posunutie textu ak používateľ rozbalí doplnkové informácie, inak by si nemusel všimnúť, že je k dispozícii ďalší text. Na vyriešenie tohto problému sme skúsili presunúť href="..." do <button>. Toto nám spôsobilo nefunkčnosť posúvania a neprešlo to ani validitou. Skúsili sme ešte triedu z <button> prehodiť do <a> a zároveň <button> odstrániť. Toto nám však nielenže zmenilo dizajn tlačidla ale tlačidlo aj znefunkčnilo. Vzhľadom k tomu, že chyba je len chybou validity a nijak nám neruší funkčnosť sme sa rozhodli ju ponechať. Posledný typ chyby bolo, že pri obrázkoch nám chýbal atribút alt. Do každého obrázka sme tento atribút doplnili.

Pri kontrole validity CSS súborov neboli nájdené žiadne chyby (Obr. 15).



Obr. 15 Validita CSS súborov

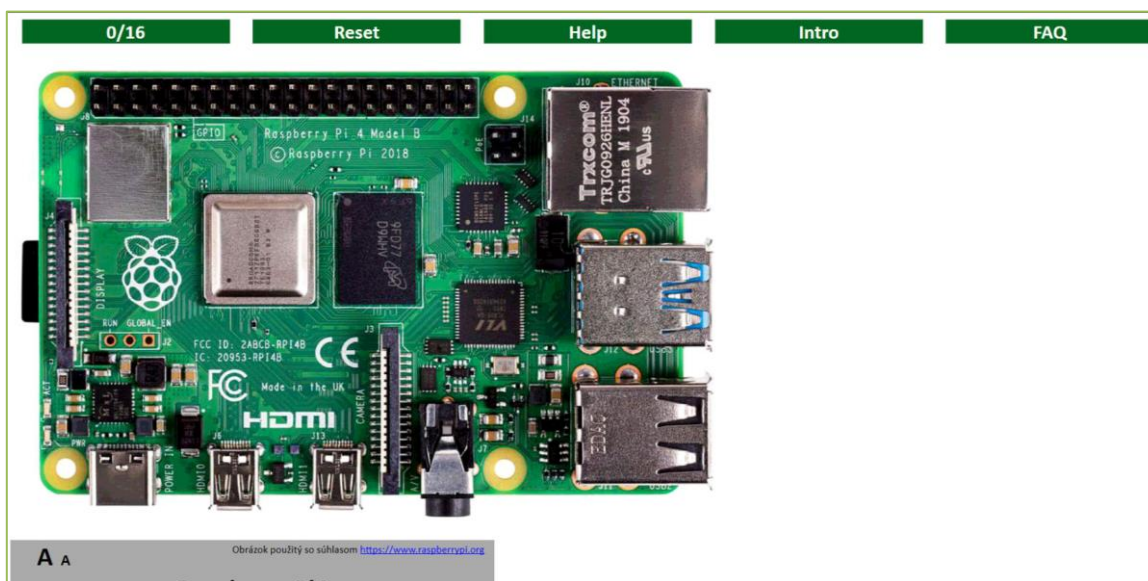
2.4.2 Testovanie funkčnosti

Našu animáciu testovalo niekoľko ľudí v rôznych prehliadačoch a s rôznymi rozlíšeniami obrazovky. Testovacia skupina sa skladala z ľudí vo veku 15 až 30 rokov. Testovanie aplikácie prebiehalo v prehliadačoch Google Chrome, Firefox, Microsoft Edge.

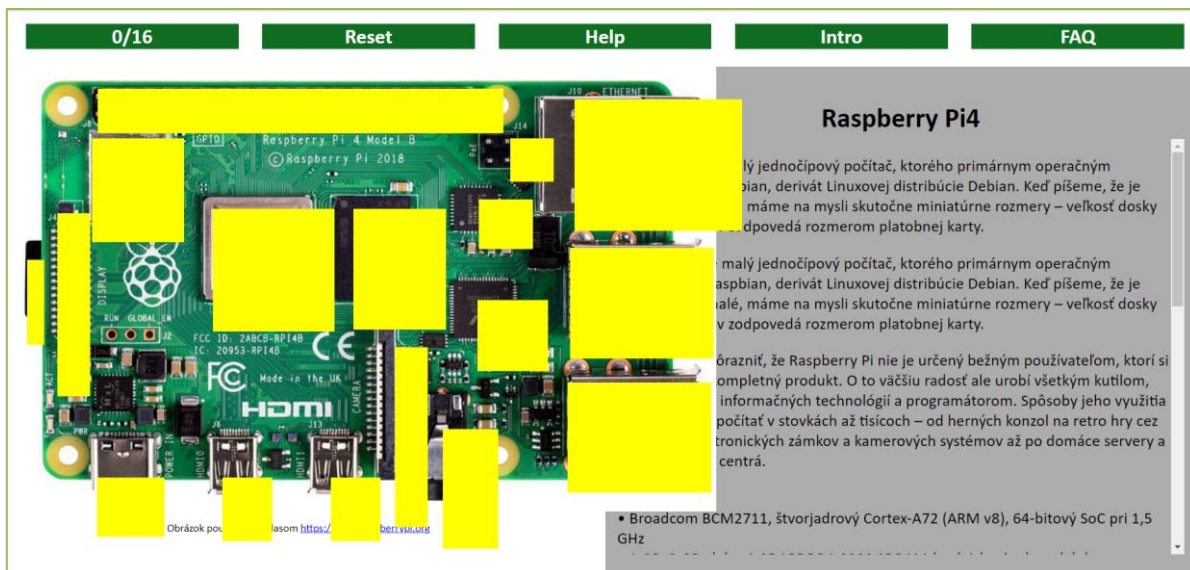
Typy prehliadačov nemali na našu animáciu žiadny vplyv. Nastali však 4 problémy. Prvým problémom bolo preblikávanie informačnej časti v hlavnej animácii po každom kliknutí na ľubovoľný komponent. Tento problém bol spôsobený CSS príkazom, ktorý určoval maximálnu možnú šírku obrázka v hlavnej časti animácie. Pri zmene animácií veľkosti obrázka a informačnej časti sa obrázok z pôvodnej veľkosti najprv zväčšil na maximálnu možnú a následne sa zmenšil na hodnotu akú mu prikázal Javascript. Tento problém sme vyriešili odstránením CSS príkazu max-width.

Druhý problém vznikol rôznymi rozlíšeniami obrazoviek. V našej práci sme využívali pixely, čo malo za následok rozhádzanie jednotlivých sekcií. Najviac sa to odzrkadlilo v hlavnej animácii, kde informačná časť nebola vedľa obrázka ale pod ním (Obr. 16) alebo boli vedľa seba a vpravo od informácií vznikol ešte veľký prázdny priestor.

Tretím problémom bolo, že ak si používateľ chcel animáciu zväčšiť alebo zmenšiť, tlačidlá komponentov sa posúvali (Obr. 17). Druhý aj tretí problém sme vyriešili tým, že miesto pixelov sme použili percentá. Týmto vzniklo skvelé zobrazenie pre všetky displeje. Zároveň sme pridali individuálny kód, ktorý zistil rozlíšenie obrazovky a ak sa animácia zobrazovala na mobiloch alebo tabletoch mala odlišné CSS.



Obr. 16 Testovanie hlavnej animácie problém zobrazenia prvkov



Obr. 17 Testovanie hlavnej časti animácie problém s tlačidlami komponentov

Menu sa zmenilo na rozklikávacie a v hlavnej časti animácie informačná časť bola pod obrázkom. Zároveň obrázok aj informačná časť boli na celú šírku displeja (Obr. 18).



Obr. 18 Zobrazenie animácie na mobile

Štvrtý problém sme si uvedomili pri hľadaní riešenia druhého a tretieho problému. V zadaní našej bakalárskej práce je podmienka objektovo orientovaného kódu, čo sme rozpísaním prvkov v HTML nespĺňali. Tento problém sa nám podarilo úspešne vyriešiť optimalizáciou kódu.

2.4.3 Zhrnutie výsledkov testov

Na základe testovania sme odhalili závažné problémy v zobrazovaní našej animácie na rôznych zariadeniach. Na zariadeniach s menšími obrazovkami aké majú napr. mobily bola naša animácia takmer nečitateľná. Na iných boli jednotlivé prvky rozhádzané. Všetky problémy sa nám však podarilo úspešne vyriešiť pomocou optimalizácie kódu a zmenou pixelov na percentá.

2.5 Optimalizácia kódu

Optimalizácia kódu je nevyhnutnou súčasťou tejto bakalárskej práce. Animácia má byť vytvorená s využitím objektovo orientovaného programovania. Objektovo orientované programovanie nám zjednodušuje vytvorenie objektov a ich aktualizáciu. V našej animácii používame až 16 objektov (1 objekt = 1 komponent v animácii).

Pri tvorbe animácie sme ich na začiatku vytvorili pomocou HTML, pridali im triedu a následne sme pomocou CSS museli každému nastavovať vlastnú, výšku, šírku, pozíciu a farbu pozadia. Toto bolo však veľmi časovo náročné a keby sme chceli kód aktualizovať, museli by sme v každom kóde hľadať jednotlivé časti a funkcie prvkov. Ďalšou nevýhodou bolo, že tento kód v programe zaberal priveľa riadkov (Obr. 19).

Keď sa pozrieme na kód lepšie, uvedomíme si, že každý objekt má rovnakú pozíciu (absolute) a rovnaké pozadie. Ďalšie veľmi podobné vlastnosti majú umiestnenie (top a left) a rozmery – výška (width) a šírka (height).

Aby sme toto nemuseli rozpisovať zakaždým, keď chceme pridať ďalší komponent vytvorili sme si konštruktor, ktorý vytvorí a zobrazí objekt. Vlastnosti ako výška, šírka a umiestnenie má každý objekt iné, preto ich posielame ako parametre (Obr. 20). Vlastnosti, ktoré majú všetky komponenty rovnaké automaticky v konšuktore pridáme bez toho aby sme ich museli posilať ako parameter. Následne sme si vytvorili premennú, ktorá zavolá tento konštruktor s konkrétnymi vlastnosťami (Obr. 21). Takto sme 17 riadkov HTML kódu a 118 riadkov kódu CSS dokázali prepísať do 35 riadkov Javascript kódu.

```

<div id="con">
  <div id="plan">
    <!--Rasperry základ-->
    

    <!--Ethernet port-->
    <div style="width:22.5%;height:27%;max-width: 22.5%;" class="eth component backCom"></div>
    <!--USB port 2.0-->
    <div style="width:19%;height:22%;max-width: 19%;" class="usb2 component backCom"></div>
    <!--USB port 3.0-->
    <div style="width:19%;height:22%;max-width: 19%;" class="usb3 component backCom"></div>
    <!--jack-->
    <div style="width:8%;height:24%;max-width: 8%;" class="jack component backCom"></div>
    <!--micro HDMI port-->
    <div style="width:7%;height:13%;max-width: 7%;" class="microHDMI HDMI1 component
    backCom"></div>
    <div style="width:7%;height:13%;max-width: 7%;" class="microHDMI HDMI2 component
    backCom"></div>
    <!--usbC-->
    <div style="width:9%;height:12%;max-width: 9%;" class="usbC component backCom "></div>
    <!--memory-->
    <div style="width:2.5%;height:17%;max-width: 2.5%;" class="memory component backCom">
    </div>
    <!--memory-->
    <div style="width:16%;height:25%;max-width: 16%;" class="procesor component backCom">
    </div>
    <!--ram-->
    <div style="width:12%;height:25%;max-width: 12%;" class="ram component backCom"></div>
    <!--DSI dispaly port-->
    <div style="width:4%;height:37%;max-width: 4%;" class="dsi component backCom"></div>
  </div>
</div>

```

```

238 }
239
240 .eth{
241   position: absolute;
242   top: 6.7%;
243   Left: 75%;
244   /*display: hide;*/
245   background: yellow;
246 }
247
248 .usb2{
249   position: absolute;
250   top: 68%;
251   Left: 79%;
252   background: yellow;
253 }
254
255 .usb3{
256   position: absolute;
257   top: 39%;
258   Left: 79%;
259   background: yellow;
260 }
261
262 .jack{
263   position: absolute;
264   top: 72.5%;
265   Left: 57%;
266   background: yellow;
267 }
268
269 .HDMI1{
270   position: absolute;
271   top: 82.5%;
272   Left: 42%;

```

Obr. 19 Vytvorenie objektov pomocou HTML a priradenie vlastností objektom pomocou CSS

```

1 // Constructor function for Person objects
2 function Component(name, width, height, top, left){
3   this.id = name;
4   this.width = width;
5   this.height = height;
6   this.top = top;
7   this.left = left;
8
9   var div = document.createElement("div");
10  div.style.width = this.width + "%";
11  div.style.height = this.height + "%";
12  div.style.background = "yellow";
13  div.style.position = "absolute";
14  div.style.top = this.top + "%";
15  div.style.left = this.left + "%";
16  div.setAttribute("class", this.id + " component backCom");
17  $("#plan").append(div);
18 }
19 }
20

```

Obr. 20 Konštruktor objektu

```

20
21 var ethernet = new Component("eth", 22.5, 27, 6.7, 75);
22 var usb2 = new Component("usb2", 19, 22, 68, 79);
23 var usb3 = new Component("usb3", 19, 22, 39, 79);
24 var jack = new Component("jack", 8, 24, 72.5, 57);
25 var hdmi1 = new Component("microHDMI HDMI1", 7, 13, 82.5, 42);
26 var hdmi2 = new Component("microHDMI HDMI2", 7, 13, 82.5, 28);
27 var usbc = new Component("usbC", 9, 12, 82.5, 11);
28 var memory = new Component("memory", 2.5, 17, 38.5, 1.7);
29 var procesor = new Component("procesor", 16, 25, 28.8, 26.6);
30 var ram = new Component("ram", 12, 25, 28.8, 45.4);
31 var dsi = new Component("dsi", 4, 37, 29.6, 5.8);
32 var csi = new Component("csi", 4, 36, 56.5, 51);
33 var blueth = new Component("blueth", 12, 21, 14.6, 10.4);
34 var usbControll = new Component("usbControll", 9.5, 14, 47, 62);
35 var ethControll = new Component("ethControll", 7.5, 11, 26, 62);
36 var poe = new Component("poe", 6, 9, 14, 66);
37 var pin = new Component("pin", 54, 9, 4, 11);

```

Obr. 21 Vytvorenie objektu

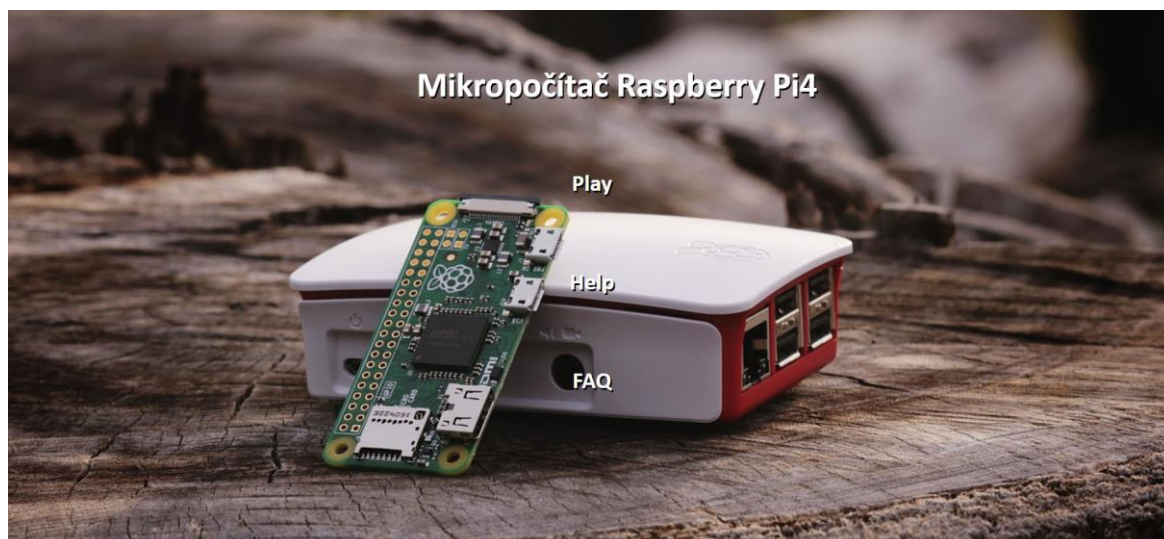
Podobne by sme mohli pokračovať aj s funkciami jednotlivých objektov. Toto vylepšenie sme už však nestihli spraviť. Napriek tomu, ak v budúcnosti k mikropočítaču pribudne nejaký komponent alebo naopak bude z neho odstránený, nebudeme musieť meniť celú animáciu ale postačí nám pridať alebo zmazať daný objekt.

2.6 Používateľská dokumentácia

Používateľská príručka je súčasťou interaktívnej animácie. Používateľ ju môže nájsť v dialógovom okne tlačidla „Help“. Cieľom používateľskej dokumentácie je vysvetliť používateľovi ovládanie animácie a všetkých jej skrytých prvkov.

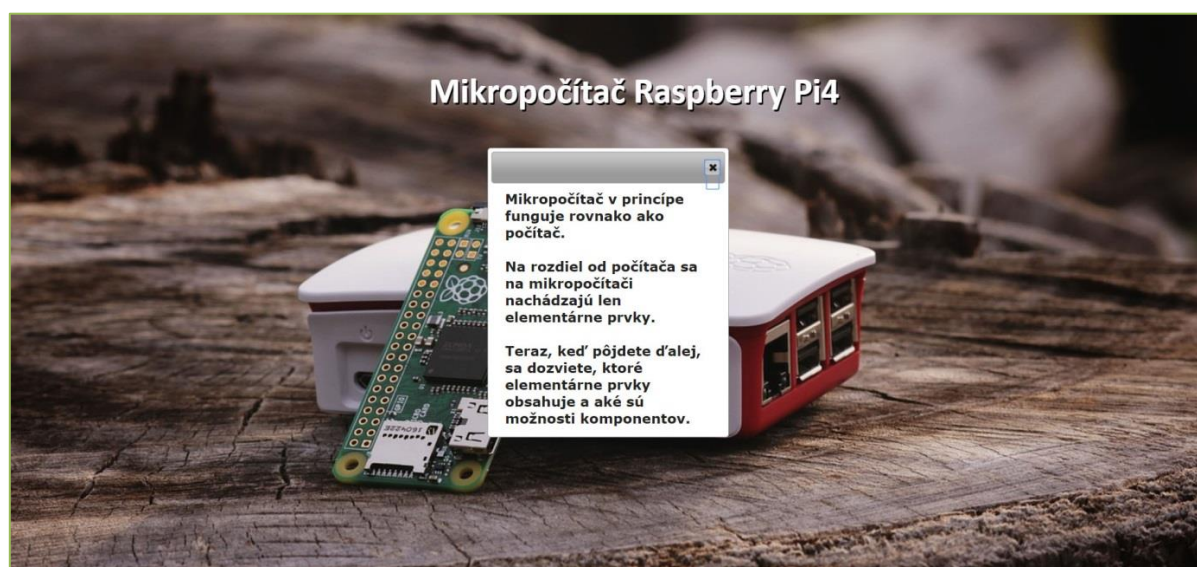
2.6.1 Intro

Po načítaní animácie (Obr. 22) sa ako prvé zobrazí názov animácie (Mikropočítač Raspberry Pi4) a ovládacie tlačidlá („Play“, „Help“, „FAQ“) s obrázkom na pozadí. Všetky obrázky v animácii sú použité so súhlasom oficiálnej stránky <https://www.raspberrypi.org>. Pomocou tlačidla „Help“ zobrazíme nápovedu. Po kliknutí na tlačidlo „FAQ“ sa dostaneme do sekcie kontrolných otázok.



Obr. 22 Úvodná stránka animácie

V názve animácie Mikropočítač Raspberry Pi4 slovo „Mikropočítač“ funguje podobne ako tlačidlo „Help“ (Obr. 23). Nepreklikne nás na ďalšiu podstránku, ale pomocou dialógového okna priblíži používateľovi základný princíp mikropočítačov. Pomocou tlačidla „Play“ sa dostaneme do hlavnej časti animácie.

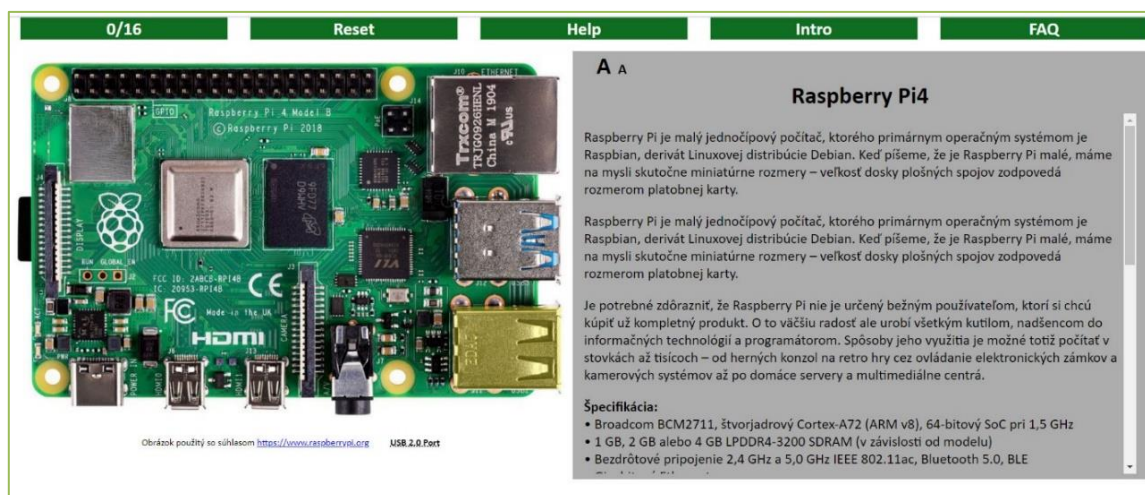


Obr. 23 Zobrazenie dialógového okna po kliknutí na tlačidlo "Mikropočítač"

2.6.2 Play

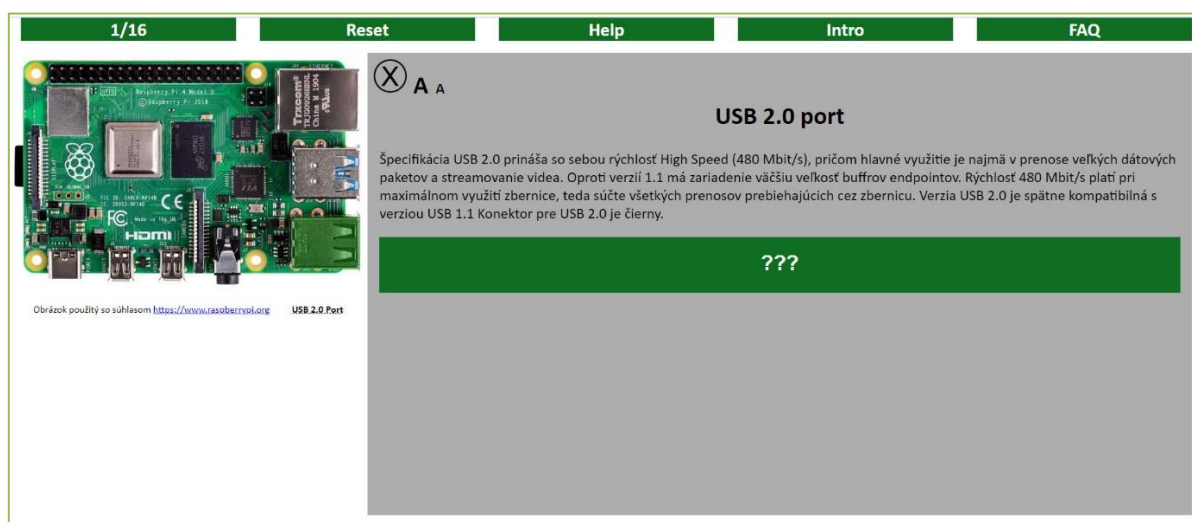
Po spustení hlavnej časti animácie uvidíme obrazovku rozdelenú na tri základne časti (Obr. 24). V hornej časti obrazovky máme menu, ktoré je okrem klasických tlačidiel „Help“, „Intro“ a „FAQ“ doplnené o tlačidlo „Reset“, ktorým si celú animáciu spustíme na novo a o políčko „0/16“, ktoré nám zobrazuje počet preštudovaných komponentov z celkového

počtu 16. Pod menu v ľavej časti vidíme zobrazený mikropočítač Raspberry Pi4. Na tomto obrázku sa nachádzajú skryté tlačidlá jednotlivých komponentov, ktoré je možné vidieť po prejdení kurzorom na daný komponent a zároveň sa pod obrázkom zobrazí informácia o tom aký komponent zobrazujeme.



Obr. 24 Zobrazenie samotnej animácie

Ak daný komponent ešte nebol bližšie preštudovaný jeho tlačidlo je vysvietené na žltó (Obr. 24). Na Obr. 25 je zobrazená animácia po kliknutí na toto tlačidlo. Farba tlačidla sa zmení na zelenú, ľavá časť, zobrazujúca mikropočítač, sa zmenší a pravá, obsahujúca informácie, sa zväčší.

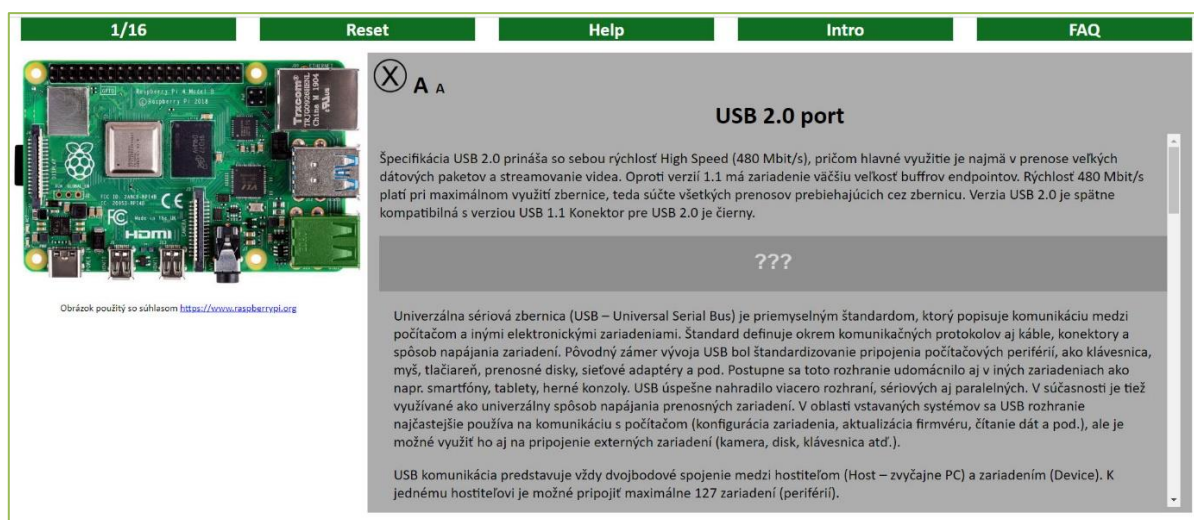


Obr. 25 Zobrazenie animácie po kliknutí na tlačidlo

V pravej časti animácie sa zobrazujú informácie o jednotlivých komponentoch mikropočítača. Tieto informácie sa menia podľa kliknutého tlačidla. Zároveň s informáciami

ostáva zobrazené aj tlačidlo, ktoré je v tejto chvíli vysvietené na zeleno a používateľovi poskytuje informáciu o tom, kde sa daný komponent na mikropočítači nachádza.

Niektoré informácie o jednotlivých komponentoch skrývajú ďalšie všeobecné informácie o komponente, ktoré môžeme zobrazíť kliknutím na tlačidlo „??“. Funkčnosť tohto tlačidla môžeme vidieť na Obr. 26.

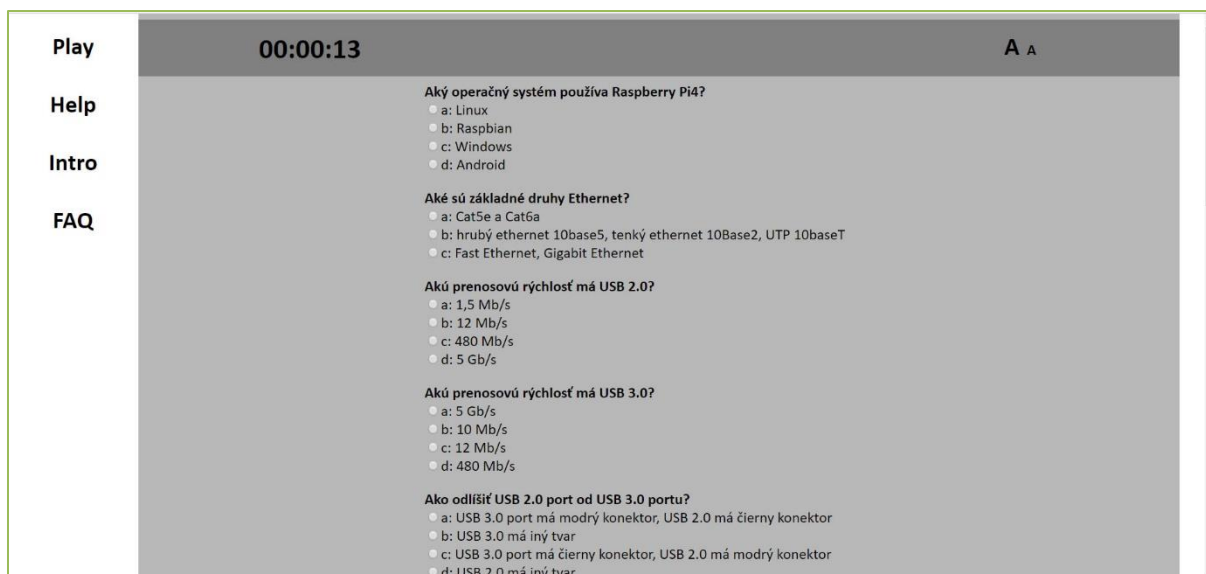


Obr. 26 Zobrazenie animácie po kliknutí na rozšírené informácie

V pravej časti animácie je možné pomocou tlačidiel „AA“ meniť veľkosť písma informácií o danom komponente. Do základného zobrazenia animácie sa dostaneme tlačidlom ⊗.

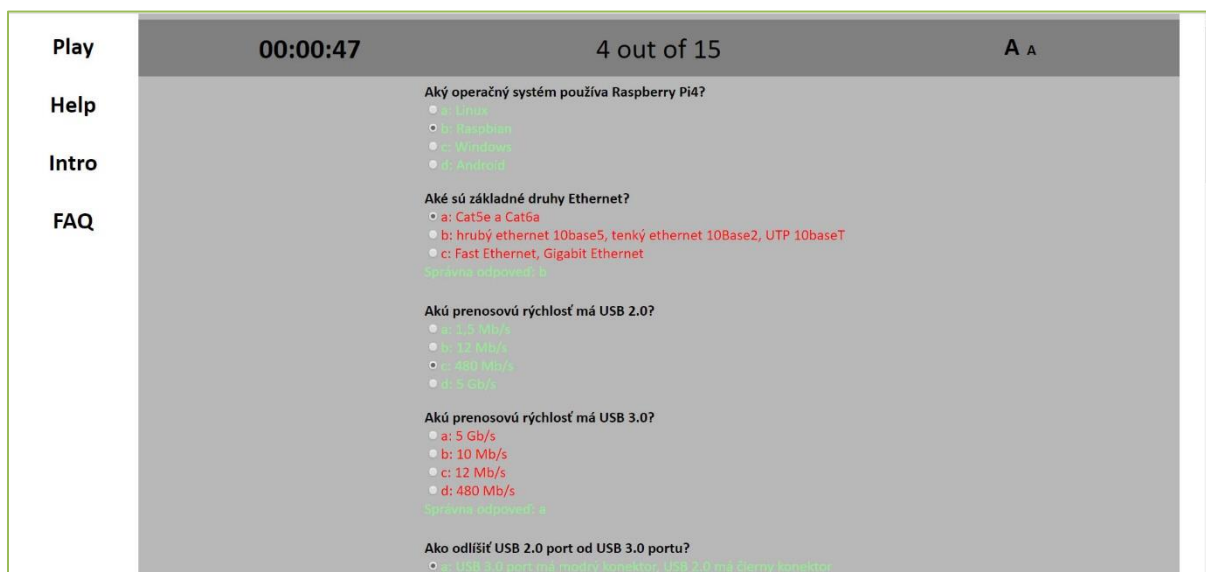
2.6.3 FAQ

FAQ je podstránka, kde si používateľ môže spraviť kontrolný test a vďaka tomu zistiť ako na tom je s vedomosťami o mikropočítači Raspberry Pi4 a jeho komponentoch. Táto podstránka je rozdelená na dve časti: menu a test (Obr. 27).



Obr. 27 Zobrazenie kontrolného testu

V ľavej hornej časti testu sa zobrazuje časomiera, ktorá sa spúšťa spolu s načítaním webstránky a zastaví sa spolu s odoslaním testu. V pravej hornej časti si znova môžeme upraviť veľkosť písma. Samotný test je vyklikávací, každá otázka má iba jednu správnu odpoveď. Po odoslaní testu sa uprostred hornej časti zobrazí počet správnych odpovedí z celkového počtu 15 (Obr. 28). Otázky sú po odoslaní vyznačené červenou alebo zelenou farbou podľa toho ako používateľ odpovedal. Ak odpovedal správne otázka sa vyznačí na zeleno. Ak odpovedal nesprávne, otázka sa vyznačí na červeno a pod otázkou sa zelenou farbou zobrazí správna odpoveď. Ak otázka nebola zodpovedaná vyhodnotí sa ako nesprávna.



Obr. 28 Zobrazenie kontrolného testu po vyhodnotení

Záver

Interaktívne animácie o technických témach majú svoje miesto nielen v rámci vzdelávacieho procesu ale plnia aj funkciu efektívneho nástroja na zvyšovanie záujmu mladých ľudí o vedu a techniku. Táto bakalárska práca sa zaoberá návrhom a vývojom interaktívnej edukačnej animácie princípu činnosti mikropočítača Raspberry Pi4 model B.

V rámci analýzy je uvedená stručná charakteristika interaktívnych animácií, analýza technológií pre vývoj týchto animácií a krátky úvod do problematiky mikropočítačov.

Postup riešenia začína návrhom ako animácia bude fungovať z programátorského a z vizuálneho hľadiska. Nasleduje implementácia animácie s opisom ako sme riešili jednotlivé situácie a prečo sme sa rozhodli práve pre toto riešenie. Opis kompletne celého kódu by bol priveľmi náročný a zdĺhavý, preto sme sa rozhodli opísať iba najpodstatnejšie časti. Aby bola vytvorená animácia plne funkčná je dôležité otestovať ju predtým ako bude publikovaná. Preto sme sa ďalej venovali testovaniu kódu. Zistili sme nedostatky animácie a následne ich v rámci optimalizácie opravili. Vďaka optimalizáciám sa zlepšila práca s animáciou, animácia sa načítava rýchlejšie a jej aktualizácia je menej náročná. Prácu na animácii sme zavŕšili vytvorením používateľskej dokumentácie, ktorá je určená študentovi pre lepšie pochopenie ovládania animácie. Animácia je používateľsky prívetivá, ale napriek tomu považujeme používateľskú príručku za nevyhnutnú súčasť, aby sme čo najviac uľahčili prácu s animáciou a zabezpečili najvyšší komfort pre používateľa. Používateľ napr. nemusí hneď vedieť, že tlačidlá sa zobrazia až po prejdení myšou nad toto tlačidlo.

Hlavným výstupom tejto práce je interaktívna animácia o princípe činnosti 16 komponentov mikropočítača Raspberry Pi4, ktorú vie používateľ plne ovládať. Používateľ si sám určuje postupnosť krokov aj rýchlosť štúdia animácie. Ako bonus pre používateľa je vytvorená možnosť otestovania si získaných znalostí v animácii testom FAQ pred spustením a po spustení animácie. Toto sa dá ľahko zneužiť. Ale táto animácia je však určená predovšetkým záujemcom o problematiku mikropočítačov a akékoľvek podvádzanie pri testovaní úspešnosti by bolo len na škodu používateľa. Vytvorená interaktívna animácia o mikropočítači Raspberry Pi4 bude voľne dostupná pre všetkých záujemcov na vzdelávacích portáloch Ústavu elektroniky a fotoniky „eLearn central“ (<http://uef.fei.stuba.sk/moodleopen/> a <http://uef.fei.stuba.sk/moodle/>).

Táto práca vznikla vďaka podpore Ministerstva školstva, vedy, výskumu a športu SR v rámci riešenia projektu KEGA 026STU-4/2019 Interaktívny showroom FINE – Fotoniky, Informatiky, Nanotechnológií a Elektroniky.

Zoznam obrázkov

Obr. 1 Mikropočítač Raspberry Pi4 model B	8
Obr. 2 UML diagram aplikácie	10
Obr. 3 Kód HTML hlavičky	12
Obr. 4 JS kód pre otvorenie dialógového okna	13
Obr. 5 Menu v úvodnej stránke	13
Obr. 6 Menu v hlavnej časti animácie	13
Obr. 7 Ľavá časť animácie - zobrazenie komponentov bez nastavenia priehľadnosti	14
Obr. 8 Ľavá časť animácie - zobrazenie komponentov až po prejdení myšou na daný komponent a zobrazenie názvu daného komponentu	15
Obr. 9 Zmena animácie po kliknutí na komponent	16
Obr. 10 Premenná obsahujúca otázky, odpovede a správnu odpoveď	17
Obr. 11 Funkcia zobrazujúca test	18
Obr. 12 Vyhodnotenie testu	19
Obr. 13 Kontrola validity HTML	19
Obr. 14 Validita podstránka "animation.html"	20
Obr. 15 Validita CSS súborov	20
Obr. 16 Testovanie hlavnej animácie problém zobrazenia prvkov	21
Obr. 17 Testovanie hlavnej časti animácie problém s tlačidlami komponentov	22
Obr. 18 Zobrazenie animácie na mobile	22
Obr. 19 Vytvorenie objektov pomocou HTML a priradenie vlastností objektom pomocou CSS	24
Obr. 20 Konštruktor objektu	24
Obr. 21 Vytvorenie objektu	25
Obr. 22 Úvodná stránka animácie	26
Obr. 23 Zobrazenie dialógového okna po kliknutí na tlačidlo "Mikropočítač"	26
Obr. 24 Zobrazenie samotnej animácie	27
Obr. 25 Zobrazenie animácie po kliknutí na tlačidlo	27
Obr. 26 Zobrazenie animácie po kliknutí na rozšírené informácie	28
Obr. 27 Zobrazenie kontrolného testu	29
Obr. 28 Zobrazenie kontrolného testu po vyhodnotení	29

Zoznam skratiek

CSS – Cascading Style Sheet, upravuje HTML

Framework – rámec, prostredie, v ktorom je napísaná aplikácia

HTML – Hypertext markup language, slúži na vytvorenie štruktúry webstránky

JavaScript – programovací jazyk

Jquery – JavaScriptová knižnica

Tag – označenie príkazu v HTML

Tutoriál - návod

Validita - platnosť, správnosť kódu

Zoznam použitej literatúry

- (1) Ako sa efektívne učiť, [online] [cit. 01.01.2020]. Dostupné z: https://cloud60.edupage.org/cloud/Ako_sa_efektivne_ucit.pdf?z%3A XuNXMbwVnvovv7mWxWDVBKM8S02QFpR9Z0mQNf1Q8lQSg%2FtuV0ehos8vXaDz9g59
- (2) Education Animations, [online] [cit. 01.01.2020]. Dostupné z: <http://institute-of-progressive-education-and-learning.org/elearning-i/elearning-educational-entertainment/education-animations/>
- (3) Interactive Animation, [online] [cit. 01.01.2020]. Dostupné z: <https://www.fudgeanimation.com/interactive-animation/>
- (4) Video and Animation, [online] [cit. 01.01.2020]. Dostupné z: <https://wearesponge.com/video-and-animation>
- (5) Objektovo orientované programovanie, [online] [cit. 18.05.2020]. Dostupné z: https://sk.wikipedia.org/wiki/Objektovo_orientovan%C3%A9_programovanie
- (6) OOP.pdf, [online] [cit. 31.05.2020]. Dostupné z: <https://www.math.sk/mpm/wp-content/uploads/2017/11/OOP.pdf>
- (7) Objektovo Orientované Programovanie v normálnej ľudskej reči, [online] [cit. 18.05.2019]. Dostupné z: <http://www.zajtra.sk/programovanie/165/objektovo-orientovane-programovanie-v-normalnej-ludskej-reci>
- (8) Počítačový expert, [online] [cit. 18.12.2019]. Dostupné z: <https://www.pocitacovyexpert.eu/co-je-to-html/>
- (9) Čo je HTML?, [online] [cit. 08.01.2020]. Dostupné z: <https://blogit.sk/co-je-html-definicia/>
- (10) Čo je HTML?, [online] [cit. 10.01.2020]. Dostupné z: <https://blogit.sk/co-je-html-definicia/>
- (11) Čo je to jazyk HTML a jeho základy. Všetko, čo na začiatok potrebujete vedieť, [online] [cit. 15.01.2020]. Dostupné z: <https://wy.sk/blog/html-zaklady/>
- (12) Kaskádové štýly [online] [cit. 15.01.2020]. Dostupné z: https://sk.wikipedia.org/wiki/Kask%C3%A1dov%C3%A9_%C5%A1t%C3%BDly
- (13) CSS – čo je to a na čo sa používa [online] [cit. 15.01.2020]. Dostupné z: <https://cookies.sk/co-je-to-css/>
- (14) When to use CSS vs. JavaScript [online] [cit. 31.05.2020]. Dostupné z: <https://gomakethings.com/when-to-use-css-vs.-javascript/>
- (15) Dedičnosť v CSS: Co to je a kterých vlastností se týká? [online] [cit. 15.01.2020]. Dostupné z: <https://www.vzhurudolu.cz/prirucka/css-dedicnost>

- (16) Selektory v jazyku CSS [online] [cit. 31.05.2020]. Dostupné z: <https://cookies.sk/selektory-v-jazyku-css/>
- (17) Aký programovací jazyk si vybrať? + Prehľad najpoužívanějších jazykov [online] [cit. 15.01.2020]. Dostupné z: <https://wy.sk/blog/programovacie-jazyky/>
- (18) JavaScript - výhody x nevýhody [online] [cit. 15.01.2020]. Dostupné z: <http://www.garth.cz/uvod-do-javascriptu/javascript-vyhody-x-nevyhody/>
- (19) JAVASCRIPT, JEHO FRAMEWORKY A AKO SA V NICH NESTRATÍ [online] [cit. 15.01.2020]. Dostupné z: <https://www.kremsa.sk/javascript-jeho-frameworky-a-ako-sa-v-nich-nestrati/>
- (20) JQuery Wikipedia [online] [cit. 05.05.2020]. Dostupné z: <https://sk.wikipedia.org/wiki/JQuery>
- (21) jQuery API [online] [cit. 05.05.2020]. Dostupné z: <https://api.jquery.com/>
- (22) jQuery je javascript knižnica novej generácie [online] [cit. 05.05.2020]. Dostupné z: <http://jquery.shaddow.sk/>
- (23) JavaScript vs JQuery – Aký je medzi nimi rozdiel? [online] [cit. 10.05.2020]. Dostupné z: <https://itvzdelavanie.wixsite.com/techweb/single-post/2017/08/13/JavaScript-vs-JQuery-%E2%80%93-Aký-je-medzi-nimi-rozdiel>
- (24) JavaScript s jQuery – lehký úvod [online] [cit. 10.05.2020]. Dostupné z: <https://www.interval.cz/clanky/javascript-s-jquery-lehky-uvod/>
- (25) Čo je Flash? [online] [cit. 06.01.2020]. Dostupné z: https://di.ics.upjs.sk/informatika_na_zs_ss/studijny_material/grafika/flash/co_je_flash.htm
- (26) Čo je to Flash alebo Web pohybuj sa [online] [cit. 06.01.2020]. Dostupné z: <https://mreferaty.aktuality.sk/co-je-to-flash-alebo-web-pohybuj-sa/referat-19175>
- (27) Čo je Flash? [online] [cit. 16.01.2020]. Dostupné z: https://di.ics.upjs.sk/informatika_na_zs_ss/studijny_material/grafika/flash/co_je_flash.htm
- (28) ActionScript [online] [cit. 12.05.2020]. Dostupné z: <https://en.wikipedia.org/wiki/ActionScript>
- (29) Flash [online] [cit. 16.01.2020]. Dostupné z: http://www.zshorakhk.cz/files/tinymce/tvorba/ucitele/flash_n/index.php
- (30) Čo je Flash? [online] [cit. 22.01.2020]. Dostupné z: https://di.ics.upjs.sk/informatika_na_zs_ss/studijny_material/grafika/flash/co_je_flash.htm

- (31) Prečo končí Adobe Flash budúci rok a ako sa následne dostať k jeho obsahu? [online] [cit. 02.02.2020]. Dostupné z: <https://vosveteit.sk/preco-konci-adobe-flash-buduci-rok-a-ako-sa-nasledne-dostat-k-jeho-obsahu/>
- (32) Aký programovací jazyk si vybrať? + Prehľad najpoužívanějších jazykov [online] [cit. 02.02.2020]. Dostupné z: <https://wy.sk/blog/programovacie-jazyky/>
- (33) Java Introduction [online] [cit. 02.02.2020]. Dostupné z: https://www.w3schools.com/java/java_intro.asp
- (34) Mikropočítač wikipedia[online] [cit. 02.02.2020]. Dostupné z: <https://sk.wikipedia.org/wiki/Mikropo%C4%8D%C3%ADta%C4%8D>
- (35) UPTON E. HALFACREE G., Raspberry Pi Uživatelská příručka, Brno: Computer Press, 2013, ISBN 9788025141168.
- (36) UPTON E. HALFACREE G., Raspberry Pi Uživatelská příručka, Brno: Computer Press, 2013, ISBN 9788025141168.
- (37) Raspberry Pi wikipedia[online] [cit. 02.02.2020]. Dostupné z: https://en.wikipedia.org/wiki/Raspberry_Pi
- (38) UPTON E. HALFACREE G., Raspberry Pi Uživatelská příručka, Brno: Computer Press, 2013, ISBN 9788025141168.
- (39) What is the Raspberry Pi 4? Everything you need to know about the tiny, low-cost computer [online] [cit. 02.02.2020]. Dostupné z: <https://www.zdnet.com/article/what-is-the-raspberry-pi-4-everything-you-need-to-know-about-the-tiny-low-cost-computer/>

Príloha

Štruktúra elektronického média

Bakalarska-Praca.pdf

Bakalarska-Praca

Bakalarska-Praca/animation.html

Bakalarska-Praca/index.html

Bakalarska-Praca/test.html

Bakalarska-Praca/uzivatelska-dokumentacia.pdf

Bakalarska-Praca/css

Bakalarska-Praca/css/style.css

Bakalarska-Praca/css/style_index.css

Bakalarska-Praca/css/style_mobile.css

Bakalarska-Praca/css/style_test.css

Bakalarska-Praca/css/style_test_mobile.css

Bakalarska-Praca/img

Bakalarska-Praca/img/AudioVideoJack.png

Bakalarska-Praca/img/classic-vs-low-energy.png

Bakalarska-Praca/img/ethernetcablepairs.png

Bakalarska-Praca/img/ethernet-chart.png

Bakalarska-Praca/img/GPIO.png

Bakalarska-Praca/img/GPIO-Pinout-Diagram-2.png

Bakalarska-Praca/img/hardware-overview-1400.jpg

Bakalarska-Praca/img/ikon-exit.png

Bakalarska-Praca/img/label.jpg

Bakalarska-Praca/img/nwordb.jpg

Bakalarska-Praca/img/pi-4-features.png

Bakalarska-Praca/img/raspberry-pi.jpg

Bakalarska-Praca/img/label.jpg

Bakalarska-Praca/img/RaspberryPi-PI4MODEL B.jpg

Bakalarska-Praca/img/USBkomunikacnyTok.jpg

Bakalarska-Praca/js

Bakalarska-Praca/js/classes.js

Bakalarska-Praca/js/dialog.js

Bakalarska-Praca/js/fontSize.js

Bakalarska-Praca/js/mobileMenu.js

Bakalarska-Praca/js/plusInfo.js

Bakalarska-Praca/js/script.js

Bakalarska-Praca/js/test.js