

**UNIVERZITA PALACKÉHO V OLOMOUCI**

**PEDAGOGICKÁ FAKULTA**

Katedra technické a informační výchovy

## **Bakalářská práce**

**Tvorba interaktivních her pro internetové prohlížeče**

**Roman Krymlák**

III. ročník – prezenční studium

Obor: Speciální pedagogika pro 2. stupeň základních škol a střední školy  
a základy technických věd a informačních technologií pro vzdělávání

**Vedoucí práce: Mgr. Jan Kubrický, Ph.D.**

OLOMOUC 2015

### **ČESTNÉ PROHLÁŠENÍ**

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a použil jen uvedených pramenů a literatury v seznamu použité literatury.

V Rapotíně dne 13. 4. 2015

.....  
Roman Krymlák

## **PODĚKOVÁNÍ**

Rád bych tímto poděkoval vedoucímu mé bakalářské práce, panu Mgr. Janu Kubrickému, Ph.D., za jeho rady, trpělivost a čas, který mi věnoval při řešení dané problematiky.

V neposlední řadě také děkuji mé rodině a přátelům, za podporu při studiu.

---

# OBSAH

---

|                                                         |           |
|---------------------------------------------------------|-----------|
| <b>OBSAH</b>                                            | <b>4</b>  |
| <b>ÚVOD</b>                                             | <b>6</b>  |
| <b>I. TEORETICKÁ ČÁST</b>                               | <b>8</b>  |
| <b>1 WEBOVÁ APLIKACE</b>                                | <b>8</b>  |
| <b>1.1 Zařazení webových aplikací</b>                   | <b>8</b>  |
| 1.1.1 Nativní, desktopové aplikace                      | 9         |
| 1.1.2 Webová aplikace                                   | 10        |
| 1.1.3 Hybridní aplikace                                 | 11        |
| <b>1.2 Princip fungování webové aplikace</b>            | <b>12</b> |
| 1.2.1 Statická webová stránka                           | 12        |
| 1.2.2 Dynamická webová stránka                          | 14        |
| <b>1.3 Možnosti tvorby</b>                              | <b>17</b> |
| 1.3.1 JavaScript                                        | 17        |
| 1.3.2 PHP a MySQL                                       | 20        |
| 1.3.3 Adobe Flash a ActionScript                        | 21        |
| 1.3.4 Jiné jazyky                                       | 23        |
| <b>2 HTML5</b>                                          | <b>24</b> |
| <b>2.1 Struktura dokumentu</b>                          | <b>25</b> |
| <b>2.2 Elementy</b>                                     | <b>27</b> |
| 2.2.1 Canvas                                            | 27        |
| 2.2.2 Audio                                             | 30        |
| 2.2.3 Video                                             | 30        |
| 2.2.4 Podpora ze strany prohlížečů                      | 31        |
| <b>2.3 WAI-ARIA</b>                                     | <b>32</b> |
| <b>3 NÁSTROJE A SOFTWARE PRO TVORBU WEBOVÉ APLIKACE</b> | <b>34</b> |
| <b>3.1 Grafické aplikace</b>                            | <b>35</b> |
| 3.1.1 Adobe Photoshop                                   | 36        |
| 3.1.2 GIMP                                              | 36        |
| <b>3.2 Editory zdrojových kódů</b>                      | <b>37</b> |
| 3.2.1 PSPad                                             | 37        |
| 3.2.2 Adobe Dreamweaver                                 | 37        |
| <b>3.3 Editace zvuku</b>                                | <b>38</b> |
| 3.3.1 Audacity                                          | 38        |
| <b>3.4 Webový prohlížeč</b>                             | <b>39</b> |
| 3.4.1 Vývojářské nástroje                               | 41        |

|            |                                            |           |
|------------|--------------------------------------------|-----------|
| <b>3.5</b> | <b>Typy her v HTML5</b>                    | <b>42</b> |
|            | Bejeweled                                  | 42        |
|            | WebQuake                                   | 42        |
|            | Robots Are People Too                      | 43        |
| <b>3.6</b> | <b>Omezení HTML5</b>                       | <b>44</b> |
|            | Chybějící funkce                           | 44        |
|            | Bezpečnost                                 | 44        |
|            | Element Audio                              | 45        |
| <b>II.</b> | <b>PRAKTICKÁ ČÁST</b>                      | <b>46</b> |
| <b>4</b>   | <b>WEBOVÁ APLIKACE PEXESO</b>              | <b>46</b> |
| <b>4.1</b> | <b>Historie a pravidla hry PEXESO</b>      | <b>46</b> |
| <b>4.2</b> | <b>Interaktivní hra PEXESO</b>             | <b>46</b> |
| 4.2.1      | Zvuk                                       | 47        |
| 4.2.2      | Prostředí hry                              | 48        |
| 4.2.3      | Programování hry                           | 49        |
| 4.2.4      | Offline                                    | 54        |
| <b>4.3</b> | <b>Komu je hra určena a její složitost</b> | <b>55</b> |
| 4.3.1      | Přístupnost a možnosti rozšíření           | 55        |
| <b>5</b>   | <b>ZÁVĚR</b>                               | <b>56</b> |
| <b>6</b>   | <b>SEZNAM ZDROJŮ</b>                       | <b>57</b> |
| <b>7</b>   | <b>SEZNAM OBRÁZKŮ A TABULEK</b>            | <b>59</b> |
| <b>8</b>   | <b>SEZNAM PŘÍLOH</b>                       | <b>60</b> |

# ÚVOD

Od doby vzniku počítačů, internetu a webových stránek samotných uplynula již poměrně dlouhá doba, během které jsme se dočkali mnohých změn. Změny se týkají nejen hardwaru, na kterém běží aplikační software, který denně používáme a který je neodmyslitelnou součástí našich životů, ale hlavně pojetí internetu, jako takového.

S internetem přímo souvisí služba WWW, která je používána širokou veřejností ze všech služeb internetu asi nejvíce. Skrze tuto službu se dostáváme k systému webových stránek v internetu, pomocí webových prohlížečů spouštěných na všech možných platformách. A právě tyto webové stránky se dočkaly mnohých a nemalých změn a každá z těchto změn posouvá možnosti tvorby a vývoje webu dále.

Jednou takovou změnou je vývoj nového standardu HTML5, kterého se ze všech stran asi nejvíce dotýká tato práce.

Došlo k využití webového prohlížeče i jinak, než jako klasického programu pro prohlížení statického obsahu, jako textu a obrázků. Využívá ho každý, kdo má počítač a připojení k internetu, nebo si ho aspoň někdy vyzkoušel. Dočkali jsme se využití webového prohlížeče, jako plnohodnotného interpretera jazyka JavaScript a jeho knihoven. Nové možnosti HTML5 utváří další kapitolu vývoje webu a jeho obsahu.

V této práci se postupně v kapitolách zaměřím na webové aplikace a jejich rozdělení, jak fungují a možnosti tvorby. Dále přiblížím nejdůležitější nové elementy standardu a jejich jednoduché použití. Poté se budu zabývat důležitými nástroji, které by uživatel mohl využít, při tvorbě jeho aplikace a omezení samotného standardu, kam až by tvorba aplikace mohla zajít. Nakonec se budu zabývat hrou, vytvořenou ve webovém prostředí s názvem PEXESO, vytvořené za pomoci elementů HTML5, JavaScriptu, kaskádových stylů, zvuku a obrázků.

## **CÍL**

Hlavním cílem práce je prozkoumat možnosti tvorby interaktivních aplikací pro internetové prohlížeče, společně se software s tvorbou souvisejícím a s využitím elementů nového standardu značkovacího jazyka HTML – HTML5.

---

Cílem praktické části této práce je vytvoření interaktivní hry na motiv karetní hry PEXESO, za použití prvků výše zmíněného standardu, kaskádových stylů, jazyka JavaScript a jeho knihoven.

# I. Teoretická část

---

## 1 WEBOVÁ APLIKACE

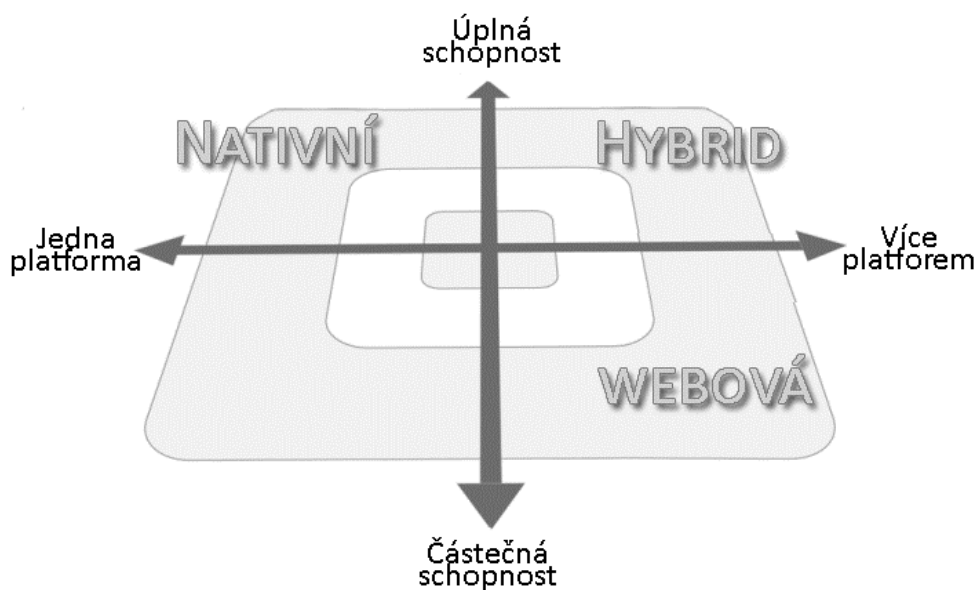
Webová aplikace je, jako každá jiná aplikace, vyvinuta pro určitou činnost. Je to jedna z architektur, které se dnes používají při vývoji mobilních aplikací. Ať se jedná o jednoduchou správu firemní databáze, nebo složitý informační systém pro školy, může být projekt řešen, právě vytvořením multiplatformní webové aplikace. Do webových aplikací můžeme dnes zařadit i hry. Jedné takové jednoduché hry se bude týkat i praktická část této práce.

### 1.1 Zařazení webových aplikací

Webové aplikace můžeme zařadit mezi existující 3 základní architektury, které se dnes využívají v prostředí osobních počítačů, mobilních telefonů a jiných mobilních zařízení s operačním systémem i bez něj (např. tablety). [1] [2]

Jsou to:

- **nativní aplikace**
- **webové aplikace**
- **hybridní aplikace**



Obrázek 1 – rozdělení jednotlivých architektur dle jejich schopností pracovat offline a závislosti na platformě (převzato a upraveno z [5])

– mobilní web

Dále bych chtěl zaměřit svoji pozornost na rozdíly webových aplikací oproti jiným architekturám, jejich výhody a jak se navzájem ovlivňují.

### **1.1.1 Nativní, desktopové aplikace**

Jsou to aplikace, které jsou vytvořené jako program pro určité zařízení např. tablet nebo i osobní počítač. Samotná nativní aplikace může ovládat určitou webovou stránku, službu (např. stravovací systém), mít svou vlastní funkci, nebo se může jednat jen o prostou hru. Název desktopové je z anglického „desktop“, což by se dalo volně přeložit jako „na stole“. Jedná se o aplikaci pro stolní počítač, ve kterém je přímo nainstalován aplikační software (desktopová aplikace – např. MS Word). [3]

Musejí být vytvořeny pro určité prostředí, platformu<sup>1</sup> (iOS, Android, Windows aj.) a hardware. Samozřejmě s použitím vývojových nástrojů a jazyka, které daná platforma podporuje (např. XCode pro iOS, Java a Eclipse pro Android). Aplikace této architektury jsou distribuovány skrz tržiště aplikací, kde jsou nabízeny k instalaci.

Tržiště je dá se říct databáze aplikací, kde si každý může najít aplikaci, jakou potřebuje. Pokud uživatel hledá nějakou aplikaci, pro řešení určitého problému, měl by zavítat právě na tržiště, neboli anglicky store. Každá platforma má své tržiště. Společnost Apple má AppStore, kde jsou nabízeny aplikace pro iPhone a iPad. Zařízení, které pracují na operačním systému Android, mají tržiště s názvem Google Play společnosti Google a nakonec zařízení s Windows mobile mají své tržiště Windows Phone Store společnosti Microsoft. [4]



*Obrázek 2 – odkazy a značky všech tří dostupných tržišť  
(Dostupné z <http://www.seznam.cz>)*

Velkým omezením je stále kapacita baterií dnešních zařízení, proto se aplikace a jejich kód optimalizují i z hlediska šetrnosti spotřeby. Jejich velkou výhodou je, že mohou pracovat i offline a tím energii šetřit.

---

<sup>1</sup> Platforma - základní počítačový nebo mobilní systém, na kterém lze aplikační software spustit

Nativní aplikace vypadají nejlépe ze všech architektur a podávají i lepší výkon, protože běží přímo na hardwaru daného zařízení, a využívají GPU<sup>2</sup> akcelerace a multithreading<sup>3</sup>. [5]

**Výhody** nativních aplikací jsou:

- využití hardwaru zařízení, ve kterém je nainstalována (kamera, GPS, dotyková obrazovka, kalendář aj.)
- mohou nabídnout větší počet funkcí
- intuitivnější ovládání, přehlednější a rychlejší uživatelské prostředí
- výkon a rychlost zpracování dat
- propagace (Google Play, App Store aj.)
- snadné použití
- automatická aktualizace

**Nevýhody:**

- obtížnější vývoj
- vývoj je třeba řešit pro různé platformy, různé operační systémy a jejich verze, různé velikosti obrazovek a funkce jednotlivých zařízení (kompatibilita). To se týká i testování, opravy i upgradu.
- chybu nelze opravit ihned
- velikost aplikace, spouštění na starších zařízeních [1]

### **1.1.2 Webová aplikace**

Tato architektura je jednodušší než architektura nativní. Jedná se o aplikaci, neboli aplikační program, který je uložen na vzdáleném serveru a který je obsluhován přes internet skrz obyčejný internetový prohlížeč. Většinou se jedná o síťovou architekturu klient-server. Kdy jsou všechna data, která jsou zpracovávána prohlížečem umístěna na webovém serveru v síti internetu.

Veškeré komponenty webové stránky, které plní pro uživatele nějakou funkci můžeme označit za webovou aplikaci. I obyčejné statické webové stránky vytvořené za pomoci jednoduchých struktur značkovacího jazyka HTML jsou webovou aplikací.

---

<sup>2</sup> GPU - grafický procesor

<sup>3</sup> Multithreading - schopnost operačního systému provádět různé části programu, tzv. vlákna současně

Existují samozřejmě i složitější, propracovanější a robustnější aplikace této architektury.

Pomocí těchto aplikací můžeme například tvořit hudbu, používat e-mailového klienta, vytvářet různé kompozice fotografií a obrázků a upravovat je, aniž bychom museli instalovat nějaký další speciální nebo složitý software do počítače nebo jiného mobilního zařízení. Webové stránky dnes totiž dokáží pracovat s dynamickými funkcemi, takže to vypadá jako by aplikace byla součástí programového vybavení počítače.

Na rozdíl od nativních aplikací, které jsou **kompilovány** pro určité zařízení a platformy, jsou webové aplikace **interpretovány**, tedy jejich kód je čten a překládán vykreslovacími jádry prohlížeče a jazyka JavaScript. Jelikož jsou webové aplikace interpretovány příslušným zařízením, je aplikace omezena výkonem daného zařízení, na kterém je spouštěna. Týká se to hlavně mobilních zařízení, u kterých je velký rozptyl v jejich výkonu. [4]

**Výhodou** je především cena vývoje a tvorby, nezávislost na platformě či operačním systému a možnost provedení změn s okamžitým efektem pro uživatele.

Hlavními **nevýhodami** mohou (ale nemusí) být menší uživatelská přívětivost, rychlost práce s daty a nedostupnost k řadě funkcí jako je fotoaparát, kalendář, seznam kontaktů a jiné interní funkce používaného zařízení. I když dnes, díky některým novým standardům, které byly relativně rychle implementovány do prohlížečů, se dá již k některým z nich přistupovat, díky rozhraní API<sup>4</sup> pro interakci s hardwarem.

Některé hranice se tedy nejen mezi architekturami začínají pomalu smazávat. [1]  
[4]

### **1.1.3 Hybridní aplikace**

Jak už název architektury napovídá, hybridní aplikace využívají výhod obou předchozích architektur. Jejich kód se stejně jako u webových aplikací píše webovými technologiemi (HTML5, CSS, JavaScript) a je funkční a kompatibilní se všemi platformami. Zároveň se však chová, jako nativní aplikace a může se distribuovat přes tržiště aplikací.

---

<sup>4</sup> API - Application Programming Interface – softwarové rozhraní pro programování aplikací

Spouští se lokálně přímo v daném zařízení, proto může být využívána offline, tedy bez přístupu k internetu. Může mít přístup i k některým interním funkcím daného zařízení (fotoaparát, kalendář a jiné).

## 1.2 Princip fungování webové aplikace

Webová aplikace je souborem statických a dynamických webových stránek.

**Statická webová stránka** se nemění, když o ni uživatel požádá, webový server odešle návštěvníkovi stránku přesně s takovou podobou, jaká je na serveru uložena.

**Dynamická webová stránka** se mění, webový server takovou stránku modifikuje před odesláním dat prohlížeči, který o ni požádal. Proměnlivý obsah stránky je důvodem, proč se takovýmto stránkám říká dynamické. [6]

### 1.2.1 Statická webová stránka

Statická webová stránka je, jak již bylo výše uvedeno, posílána z webového serveru směrem k uživateli, který o ni požádal skrz svůj internetový prohlížeč. A to buď kliknutím na nějaký odkaz na stránce, nebo zadáním přímé URL<sup>5</sup> adresy do textového pole pro zadávání adres, která odkazuje na daný HTML soubor.

**Stránka je tedy zasílána tak, jak ji autor této stránky vytvořil předtím, než ji uložil na webový server.** A protože se HTML po uložení na server dále nemění, říkáme takovéto stránce stránka statická.

Nemusí se však jednat o neměnitelný a stálý obsah. V tomto ohledu je pojem statický zavádějící. V tomto smyslu se jím myslí to, že je stránka zasílána uživateli beze změn.

Součástí statického obsahu může být např. flash animace (swf), nebo přechodový efekt, který stránku oživí a uživateli zpříjemní procházení webu. Dále jde takovou stránku vybavit JavaScriptem, ale o tom až dále.

Na obrázku 3 je zjednodušené schéma komunikace mezi webovým prohlížečem a serverem. Tato komunikace má několik kroků. Jsou to:

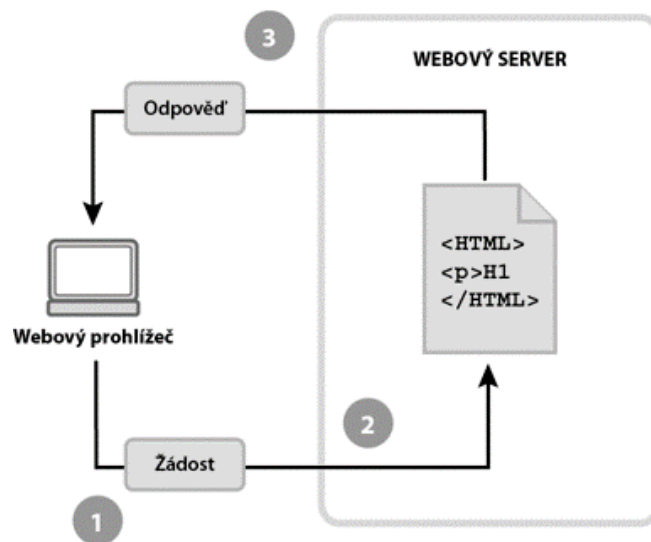
- 1) Webový prohlížeč požádá server o statickou stránku
- 2) Webový server tuto stránku vyhledá

---

<sup>5</sup>URL (UniformResourceLocator) – globální adresa dokumentů a jiných zdrojů na internetu

3) Webový server pošle tuto stránku prohlížeči, který o ni požádal [6]

Příklad jednoduchého kódu odeslaného serverem prohlížeči (dle standardu HTML5):



Obrázek 3 – schéma komunikace mezi webovým prohlížečem a webovým serverem (převzato z [6])

```
index.html
<!DOCTYPE html>
<html>
  <head>
    <title>Moje první stránka</title>
  </head>
  <body>
    <h1>Můj první nadpis</h1>
    <p>Můj první text</p>
  </body>
</html>
```

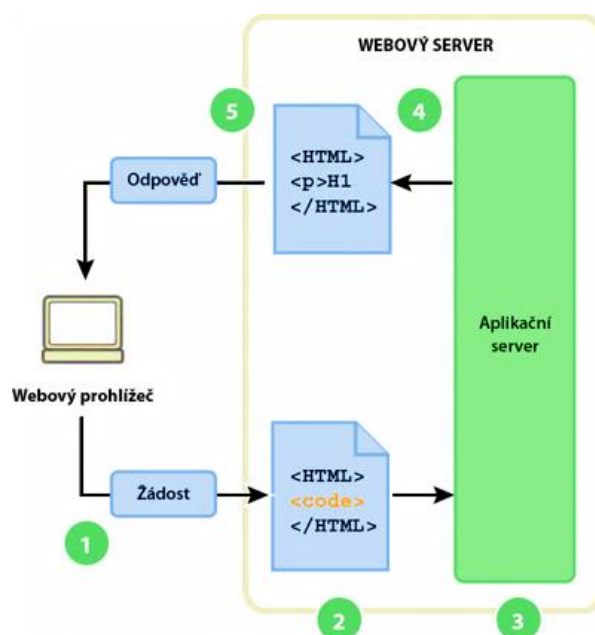
Kód zobrazí stránku s titulkem „Moje první stránka“, nadpisem první úrovně „Můj první nadpis“ a odstavcovým textem „Můj první text“.

Takový zdrojový kód si můžeme zobrazit téměř v každém prohlížeči na každé webové stránce. U většiny prohlížečů je k dispozici tato funkce v hlavní nabídce prohlížeče, kde se nachází v podnabídce s názvem „vývojové nástroje“.

### 1.2.2 Dynamická webová stránka

Druhým typem webových stránek jsou stránky dynamické. Tyto stránky obsahují tzv. „server-side“ kód, neboli **serverové skriptování**, které umožňuje serveru vytvářet unikátní obsah pro danou stránku.

**Rozdíl** oproti statickým stránkám je v tom, že dynamické stránky se na serveru zpracují dle požadavku uživatele a poté je dotazujícímu odeslána interpretovaná (interpretem daného jazyka) a vygenerovaná HTML stránka. Uživatel na funkci nic nepoznává. Zobrazují se stejně, pouze fáze zpracování je jiná.



Obrázek 4 – schéma komunikace mezi webovým prohlížečem a webovým serverem zpracovávajícím dynamickou stránku (převzato z [6])

Z přípony souboru webové stránky, lze zhruba rozpoznat, zdali se jedná o statickou, nebo dynamickou stránku. Pokud má stránka příponu „.htm“, nebo „.html“, jedná se pravděpodobně o stránku statickou. Dynamické mají potom „.php“, „.asp“, „.jsp“, i když ne všechny dynamické stránky v sobě opravdu skrývají dynamický obsah. Většina pak alespoň částečně využívá výhody serverového skriptování, alespoň co se týče layoutu<sup>6</sup> stránky.

Existují různé programovací, skriptovací jazyky (např. PHP, ASP, JSP), které se interpretují na straně serveru. Koncovky jejich souborů jsou napsány výše.

<sup>6</sup> Layout – vzhled webové stránky

Na Obrázek 4 je vidět zjednodušené schéma komunikace mezi webovým prohlížečem a serverem, který zpracovává dynamickou stránku.

Jednotlivé fáze tohoto procesu jsou:

- 1) Webový prohlížeč požádá o dynamickou stránku (.php apod.)
- 2) Webový server tuto stránku vyhledá a předá ji aplikačnímu serveru
- 3) Aplikační server najde na stránce instrukce, které provede a dokončí ji
- 4) Aplikační server předá dokončenou stránku zpět webovému serveru
- 5) Webový server pošle tuto dokončenou stránku prohlížeči, který o ni požádal [6]

Dynamické webové stránky a aplikace jimi vytvořené mohou využívat a většinou také využívají databázi, kde může být uloženo velké množství dat. Většinou se jedná, o velmi známou relační databázi MySQL. Na Obrázek 5 je příklad komunikace a zpracování dynamické stránky s použitím databáze.

Pro představu uvedu jednoduchý praktický příklad, jak funguje dynamická stránka. Na Obrázek 6 je jednoduchý formulář (statická webová stránka) „ahoj.html“, který uživatel (prohlížeč) získal požadavkem (URL adresou) od webového serveru.

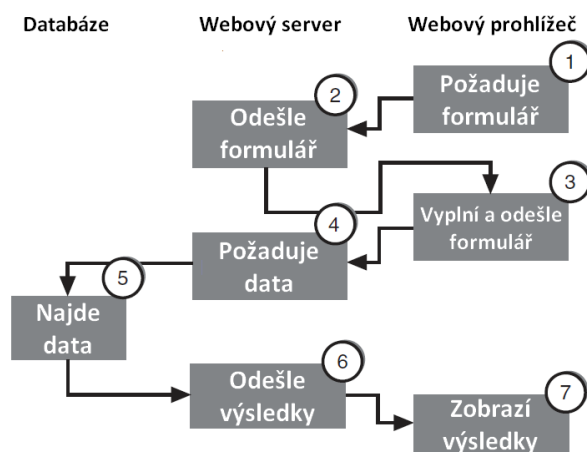


*Obrázek 5 – příklad komunikace mezi internetovým prohlížečem, webovým a databázovým serverem (převzato a upraveno z [21])*

Uživatel do něj vepíše nějaké informace. V tomto případě je vyzván k vepsání pozdravu. Vepíše pozdrav do připraveného textového pole a kliknutím na tlačítko „Odešli“ jsem se dostal k bodu č. 4 na Obrázek 5, kde se vepsaná data odevzdají např. PHP skriptu (nebo jinému z mnoha „server-side“<sup>7</sup> zpracovávaných jazyků).

---

<sup>7</sup> Server-side – skripty a programy zpracovávané nebo interpretované na serveru



Obrázek 6 – jednoduchý formulář  
v internetovém prohlížeči (Zdroj: autor)

Server (většinou Apache nebo Microsoft IIS) tato data zpracuje a interpretuje podle instrukcí mu předaných a odešle prohlížeči výsledek ve formátu prosté HTML stránky.

Níže můžete vidět kód PHP skriptu (serverový skript), kterému je obsah formuláře předáván.

```

skript.php
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        //atribut charset určuje kódování znaků pro dokument HTML
        <title>Pozdrav</title>
    </head>
    <body>
    <?php
        if (isset($_GET['pozdrav'])) {
            //zjišťuje, zda existuje superglobální8 proměnná $_GET a zda není hodnoty
            NULL
            $pozdrav=$_GET['pozdrav'];
            echo "<p>Zdravím Tě - ".$pozdrav."</p>". "\n";
        }
    ?>
    </body>
</html>

```

Existuje ještě superglobální proměnná \$\_POST, kde její obsah není vidět v adresním řádku prohlížeče.

<sup>8</sup> Superglobální proměnná – možnost přistupovat k takovéto proměnné z jakékoliv části kódu v programu

Takto vypadá výsledek kódu v prohlížeči, vrácený zpracovaným skriptem, kde v hlavičce vidíme předávanou proměnnou \$\_GET (tj. je vidět v adresním řádku prohlížeče):

```
skript.php?pozdrav=Ahoj
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Pozdrav</title>
  </head>
  <body>
    <p>Zdravím Tě - Ahoj</p>
  </body>
</html>
```

Do takového kódu se dá implementovat komunikace s již zmíněnou databází, ale to už je jiná oblast, kterou se dále podrobněji zabývat v této práci nebudu. Šlo mi jen o přiblížení a pochopení funkce a struktury mezi těmito dvěma používanými formami tvorby webových stránek a tudíž i webových aplikací.

## 1.3 Možnosti tvorby

Výše jsem rozebral a porovnal základní dvě formy tvorby, ve kterých se dají tvořit webové stránky a v nich webové aplikace. Uvedu tedy dále, jaké máme další možnosti a co můžeme k tvorbě webové aplikace využít. Podrobněji se zaměřím jen na nejpoužívanější jazyky a technologie. Existuje jich totiž celá řada a každý si může vybrat, v čem bude tvořit webovou, nebo i jinou aplikaci.

### 1.3.1 JavaScript

Důležitou součástí webových aplikací, jejichž kód je zpracováván na straně prohlížeče, je právě programovací jazyk JavaScript. Vytvořila jej společnost Netscape na konci minulého století.

Oficiální název JavaScriptového standardu je ECMA-262. Název je často zkracován zkratkou JS. JavaScript (dále tedy jen JS) nemá nic společného s programovacím jazykem Java kromě podobné syntaxe.

Jedná se o malý, lehký, interpretovaný, multiplatformní a objektově-orientovaný jazyk, který je nejvíce známý jako skriptovací jazyk, právě pro webové stránky, ale používá se i v prostředích a aplikacích mimo prohlížeče (např. Adobe Acrobat).

Nelze jej použít samostatně, ale je určen pro snadné zabudování do jiných produktů a aplikací. Např. do zmíněných webových stránek, kde může ovládat nebo upravovat jejich elementy. [7]

Jádro JS obsahuje základní sadu objektů, jako je např. pole, datum, math a jiné. Dále stejně jako jiné programovací jazyky základní sadu jazykových prvků, kterými jsou operátory, řídicí struktury a příkazy. [8]

Existují dvě možnosti, jak může být JS spouštěn a to:

- **JS na straně klienta** – rozšiřuje základní jazyk tím, že poskytuje objekty k ovládání webového prohlížeče a jeho DOM<sup>9</sup>. Například dovoluje aplikaci rozmístit, přistupovat a editovat prvky v HTML dokumentu a reaguje na různé události (kliknutí myši, zapisování do formuláře, navigace na stránce). Takto se JS používá nejčastěji, kde je skript zpracován interpretem samotného webového prohlížeče po stažení z webového serveru nebo spuštěním z lokálního úložiště.
- **JS na straně serveru** – poskytuje objekty, které jsou důležité pro běh skriptu na straně serveru. Například rozšíření umožňující aplikaci komunikovat s databází. [7]

JS a jeho knihovny jsou velice důležité prvky, pro komunikaci s elementy nového standardu HTML5 a CSS3<sup>10</sup>. Hojně se jich využívá a určitě využívat bude, čím dál častěji.

Použití samotného JS, by bylo v některých situacích velice složité pro vývoj webové aplikace. **Pro zjednodušení tvorby, přehlednosti kódu a jednoduchému přístupu k DOM elementům**, byla vyvinuta dnes asi nejpoužívanější JS knihovna jQuery.

### ***jQuery***

Je to malá, rychlá a na funkce bohatá knihovna<sup>11</sup> JS. Zjednodušuje a rozšiřuje mnoho činností, které můžeme aplikovat na HTML dokument (zpracování událostí, animace aj.).

---

<sup>9</sup> DOM – Document Object Model – platformě a jazykově neutrální rozhraní umožňující programům a skriptům dynamicky přistupovat a aktualizovat obsah, strukturu a styl dokumentu

<sup>10</sup> CSS3 – třetí verze kaskádových stylů tvořící layout stránky

Knihovna jQuery je velmi široce používanou JS knihovnou. Jedná se o knihovnu s otevřeným zdrojovým kódem. Pracuje tedy na ní celá řada dobrovolníků a programátorů.

Základním posláním této knihovny je **odstranit rozdíly mezi implementacemi v různých webových prohlížečích**. Velice těžké je vytvořit takový skript, který by ve všech prohlížečích, které se standardně používají, fungoval stejně. Kolikrát je to až nemožné toho dosáhnout.

Knihovna jQuery nezavádí objektově-orientovaný způsob programování v JS, ale používá programování, které je postaveno na událostech. Snaží se také o minimalizaci svého kódu principem řetězení volání svých metod. To co by se normálně programovalo v JS na deset řádků, stačí knihovně jQuery řádek jediný a co je však nejdůležitější, kód bude (v převážné většině) fungovat ve všech používaných prohlížečích stejně. [9]

Díky kombinaci univerzálnosti knihovny a jejího rozšíření mezi tvůrci i zasvěcenými uživateli, je dnes téměř nepostradatelnou součástí každé HTML stránky, která ke své funkci využívá JS.

### **Modernizr**

Je další JS knihovnou, která **detekuje ve webovém prohlížeči podporované funkce HTML5 a CSS3**. Jedná se o knihovnu s otevřeným zdrojovým kódem, může ji tedy rozšiřovat celá řada dobrovolníků a programátorů stejně jako u knihovny jQuery. A právě díky tomu, můžeme dosáhnout různých úrovní funkcionality v cílovém webovém prohlížeči.

Uživatel má na výběr z několika produktů, co se webového prohlížeče týče. Škála dnes používaných prohlížečů je opravdu velká, nehledě na to, že s postupným vývojem existuje i několik desítek verzí, které podporují, nebo (což je daleko horší), nepodporují různé funkce, které potřebujeme k vytváření, nebo vylepšení aplikace.

Optimalizace je někdy až nemožná, proto vznikla knihovna JS Modernizr. Umožňuje nám tedy využít veškeré nové funkce HTML5 a CSS3 a nemusíme se bát, že se nám naše aplikace nebo web, bude chovat nestandardně ve starších nebo jiných prohlížečích. [4]

---

<sup>11</sup> Knihovna – naprog. kód, který je zaměřen na zjednodušení syntaxe. Zrychluje a zpřesňuje, vytváření programovacího kódu a jeho kompatibilitu.

Je nutné zvážit, zda je pro projekt důležité a skutečně nezbytné přidávat další kB kódu a tím zpomalovat celkový chod aplikace nebo náklady na její provoz. Hlavně co se mobilních zařízení a jejich datového připojení týče.

Můžeme si tak dnes již vybrat, které funkce budeme potřebovat testovat a vygenerovat si tak vlastní část Modernizr knihovny a tím zmenšit její velikost.

V mé praktické části této bakalářské práce (hry Pexeso) knihovnu Modernizr nevyužívám, i když by se mohlo jednat o určité vylepšení a zlepšení kompatibility s dalšími prohlížeči, kde by určité nepodporované funkce mohli být nahrazeny knihovnami 3. stran a tím „obejít“ standardní fungování vloženého elementu.

Jak knihovnu implementovat a příklad použití uvedu v kapitole „*Co bude uživatel potřebovat pro tvorbu aplikace*“, jako právě možnost vylepšení aplikace.

### 1.3.2 PHP a MySQL

Další možností tvorby webové stránky či aplikace je využití dynamicky vytvářených stránek za pomoci skriptu na straně serveru. Lehce jsem se tohoto tématu a jazyka PHP dotkl v bodě o dynamických stránkách. Trochu si jej tedy přiblížíme.

PHP je nejrozšířenějším serverovým skriptovacím jazykem, speciálně vyvinutým pro web. Běží asi na 75% všech serverů.

PHP soubor má koncovku „.php“. Syntax jazyka je podobná známému a používanému jazyku C. Takže pokud uživatel ovládá tento jazyk, přechod k PHP by mu neměl dělat sebemenší problém.

Jednoduchý příklad skriptu:

|                                                                    |     |
|--------------------------------------------------------------------|-----|
|                                                                    | PHP |
| <code>&lt;?php echo "&lt;p&gt;Ahoj světe!&lt;/p&gt;"; ?&gt;</code> |     |

PHP kód začíná otevíracím tagem „<?php“ a ukončuje se zavíracím tagem „?>“. PHP skript je většinou včleněn přímo do HTML kódu, kde plní nějakou funkci.

Velice dobře se dá PHP skriptu využít například při zjednodušení tvorby layoutu stránky, kde je každá sekce (hlavička, patička, menu aj.) uložena zvlášť do samostatného PHP souboru a poté vkládána (includována) do jednotlivých stránek tvořících web.

S PHP úzce souvisí i databázový systém MySQL. Jedná se o nejpoužívanější databázi v internetu. Je malá, rychlá a speciálně vytvořená pro webové stránky. Uchovává informace např. o uživateli, výrobcích a jiná potřebná data.

MySQL se skládá ze serveru MySQL, několika obslužných programů, usnadňujících správu a administraci databáze a jiného podpůrného software.

Využívá jazyka SQL, což je standardní jazyk pro přístup k databázím a získání jejich dat (nejen MySQL). [10]

Příklad kódu, který vybere všechny záznamy v tabulce zákazníci:

| SQL                                   |
|---------------------------------------|
| <code>SELECT * FROM zakaznici;</code> |

**PHP a MySQL jsou důležitými a nepostradatelnými nástroji při tvorbě webových aplikací.** Navíc se dají kombinovat s JS, což je dělá silným nástrojem pro tvorbu webu. Tyto dva jazyky se používají i u redakčních a publikačních systémů jako je WordPress, Drupal, Joomla a jiné.

Naprogramované rozhraní v PHP a údaje v databázi, zjednodušují obyčejným uživatelům, přidávání obsahu do jejich stránek, vybraného redakčního systému.

V praktické části tuto technologii využívat nebudu, ale zmínil jsem její existenci, jako jednu z alternativ tvorby webové aplikace, než jakou využívám ve své aplikaci v závěru této práce.

Je možné vytvořit webovou aplikaci i za pomoci skriptu, který není zpracováván u klienta v jeho webovém prohlížeči, ale na straně serveru, který pouze odesílá výsledky webovému prohlížeči. Popřípadě zkombinovat obě tyto možnosti.

### **1.3.3 Adobe Flash a ActionScript**

Speciálním typem vývojového nástroje pro webové aplikace, je Adobe Flash (dříve Macromedia). Jedná se o program, ve kterém můžeme vytvářet vektorové animace a obrázky a zároveň jim přiřazovat různé akce a události. Důležitou součástí tohoto programu je programovací jazyk ActionScript. Dnes ve verzi 3.0.

Pro spuštění flash aplikace v prohlížeči potřebujeme zásuvný modul, který se nazývá – Adobe Flash Player.

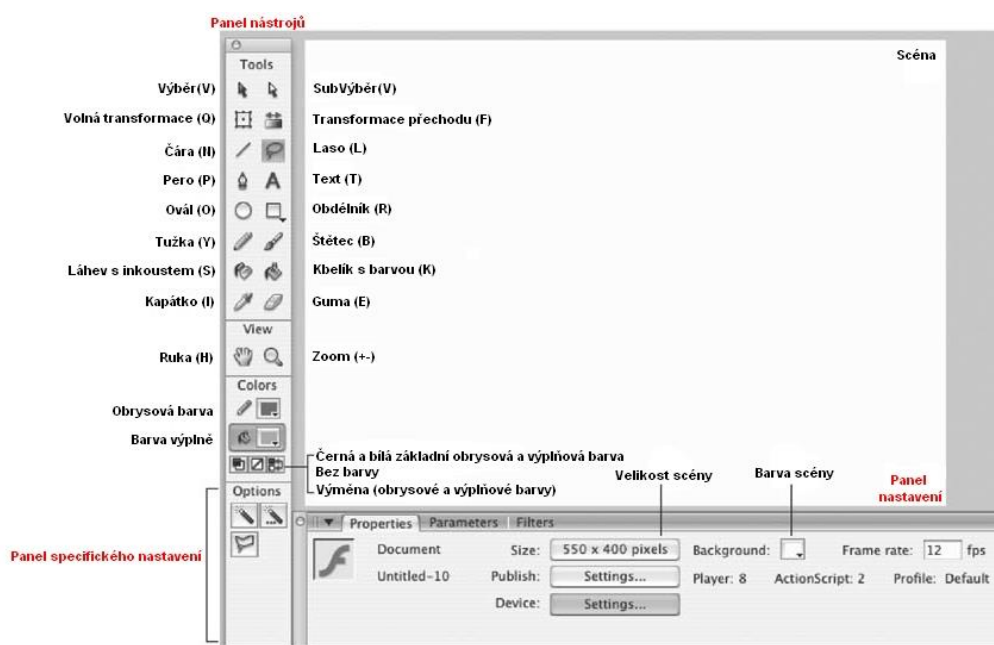
Flash je nástroj, který designéři a vývojáři využívají k tvorbě prezentací, aplikací a jiného interaktivního obsahu. Projekty mohou být: jednoduché animace, video, aplikace, hry atd.

Lze jím vytvořit bohaté Flash aplikace včetně obrázků, zvuku, videa a různých dalších věcí včetně filtrů a modifikačních nástrojů.

Flash je ideální pro projekty, které chceme sdílet prostřednictvím internetu, protože jeho soubory mají velmi malou velikost – pracuje s vektorovou grafikou.

Přímým konkurentem je právě HTML5 a jeho nové elementy. Byl i důvodem, proč a jakým způsobem byl nový standard tvořen a čím vším vylepšen.

Přímo s grafickými elementy souvisí i jazyk ActionScript (dále AS).



Obrázek 7 – vývojové prostředí Macromedia Flash 7 (Zdroj: autor)

Byl vyvinut v roce 2000 z ECMA skriptu. Patří k objektově orientovaným programovacím jazykům. Je to programovací jazyk pro aplikace, vyvíjené pomocí Adobe Flash, nebo dalších vývojářských nástrojů, využívajících stejného datového formátu. Pomocí AS se dají vytvářet internetové aplikace, animace, webové stránky a nakonec i hry. Na obrázku 7 můžete vidět starší vývojové prostředí pro Flash (dříve ještě od společnosti Macromedia).

#### **1.3.4 Jiné jazyky**

Dále se dají využít pro tvorbu webových aplikací jazyky a technologie ASP (podobné PHP), ASP.NET (neplést s ASP – nemají nic společného), XML, Python, Ruby a jiné.

Možností je určitě hodně a každý si může vybrat, ve kterém jazyce, nebo jakým způsobem bude svou aplikaci vyvíjet.

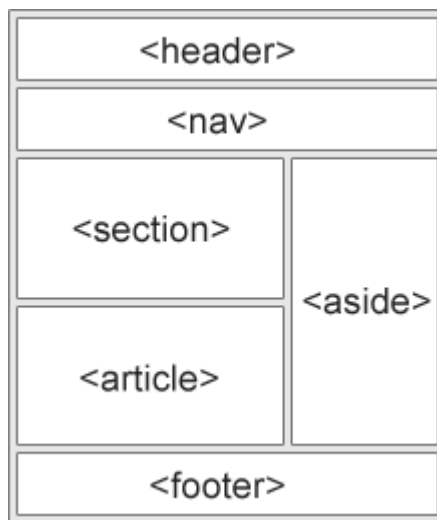
## 2 HTML5

V literatuře a na webových stránkách konsorcia W3C<sup>12</sup>, ze kterých jsem čerpal, se název standardu HTML, píše dohromady i s číslem verze (HTML5), proto se tohoto zápisu budu držet i v této práci.

Jedná se o nejnovější standard pro jazyk HTML. Předchozí verze HTML, tedy HTML 4.01 přišla v roce 1999 a od té doby se internet a webové stránky výrazně změnily.

Přišel tedy nový standard připravený a vyladěný pro tvorbu webových stránek a skutečných webových aplikací. Vylepšený o nové prvky a jejich API: lokální úložiště a databáze, 2D/3D grafika, podpora režimu offline, elementy audio a video, sockety, vlákna a jiné. [11]

Nový standard se také snaží oddělit obsah od vzhledu, proto došlo také k odstranění elementů, které byly navíc, a které byly dávno nahrazeny kaskádovými styly. Jsou to např.: font, center, strike, frame, frameset, big a další. [12]



Obrázek 8 – rozložení stránky pomocí nových elementů  
(Dostupné z: w3schools.com)

Došlo také k přidání dalších nových sémantických elementů.

---

<sup>12</sup> <http://www.w3.org/TR/html5/>

Stránku můžeme vytvořit pomocí nesémantických elementů (<div>, <span>), které nám nic neříkají o jejich obsahu nebo naopak element např. <table>, je elementem sémantickým. Říká nám, že obsahem bude jistě nějaká tabulka.

Vyhledávač Google zjistil, že se na většině webových stránek, vyskytuje v elementu div identifikátor např. „header“ (česky hlavička nebo záhlaví) a „footer“ (zápatí). Takže vznikly nové elementy: header, footer, section a jiné. Na všechny lze aplikovat kaskádové styly.

Nejde v tomto případě o nějakou závratnou novinku, ale určitě zjednoduší návrh stránek, zpřehlední zdrojový kód a samotnou indexaci vyhledávacími roboty.

Dne 28.10.2014 oznámilo konsorcium W3C dokončení a doporučení standardu HTML5. Tedy, že je standard kompletní. [13]

## 2.1 Struktura dokumentu

Abychom mohli určit rozdíly oproti předchozí verzi HTML, uvedu zde obě varianty a rozdíly mezi nimi. Patrné budou určitě na první pohled.

```
HTML 4.01
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <link type="text/css" rel="stylesheet" href="styl.css">
    <link type="text/javascript" src="javascript.js">
    <title>Dokument HTML 4.01</title>
  </head>
  <body>
    <h1>Nadpis</h1>
    <p>Text</p>
  </body>
</html>
```

```
HTML5
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="styl.css">
    <script src="javascript.js"></script>
    <title>Dokument HTML5</title>
  <body>
    <h1>Nadpis</h1>
    <p>Text</p>
  </body>
</head>
</html>
```

První změnou je deklarace doctype, uvedená na prvním řádku HTML dokumentu. Říká prohlížeči, o jaký typ dokumentu se jedná. Nejedná se tedy o tag jazyka HTML, ale pouze o informaci pro prohlížeč, dle jaké verze HTML má dokument vykreslit.

Předchozí verze HTML 4.01 byla založená na SGML<sup>13</sup>, a proto musela obsahovat i odkaz na DTD<sup>14</sup>. HTML5 již není založen na SGML, proto nevyžaduje odkaz na DTD a kód je tudíž kratší. Dle konsorcia W3C, bude každá další následující verze HTML pouze tohoto krátkého zápisu. [14]

Tento zápis je i jednodušší na zapamatování, u dřívějších verzí se špatně pamatovala celá deklarace, včetně odkazu a mnoho lidí ani nevědělo, která je vlastně ta správná. Ve většině případů docházelo k vyhledávání na internetu správné deklarace, nebo její kopírování z již vytvořených stránek. Už tedy na začátku HTML5 dokumentu vidíme vylepšení.

Deklarace doctype je důležitá i z hlediska kontroly validity stránky, čili její správnosti z hlediska správné syntaxe dle standardu. [11]

Co se týče další změny, tak tou je změna zápisu meta tagu charset, neboli kódování. Slouží k určení kódování znaků pro HTML dokument.

Prosazovaným standardem je kódování UTF-8, neboli Unicode. Jedná se o moderní znakovou sadu, která má podporu pro mnoho abeced včetně české.

Zde jsou uvedeny oba zápisy meta tagu. Jak dříve používaný, tak i nový.

| Kódování                                                                           |
|------------------------------------------------------------------------------------|
| //HTML 4.01<br><meta http-equiv="content-type" content="text/html; charset=UTF-8"> |
| //HTML5<br><meta charset="utf-8">                                                  |

Opět jako v předchozím případě došlo ke zjednodušení. Z hlediska kompatibility, by všechny prohlížeče měly tomuto zkrácenému zápisu meta tagu rozumět. Samozřejmě se tento nový zápis i lépe pamatuje. Není tedy třeba jej někde dohledávat, abychom jej zapsali správně.

Vylepšení se dočkala i deklarace elementu <link> a <script>. Změna se týká externích souborů kaskádových stylů i souborů jazyka JavaScript. Ze zápisu zmizel

---

<sup>13</sup> SGML – (Standard Generalized Markup Language) je standardní jazyk určený k formálnímu popisu struktury dokumentů

<sup>14</sup> DTD - určuje pravidla pro značkovací jazyk HTML, prohlížeče potom vykreslují obsah správně. Definuje syntaxi

atribut „type“. Kaskádové styly byly prohlášeny za standard, normu a výchozí styl pro HTML5. Stejně tomu je i u JS, jako výchozího skriptovacího jazyka. [11]

Takto upravenou stránku si můžeme nechat zkontrolovat validátorem. Například validátorem od konsorcia W3C<sup>15</sup>. Tam zjistíme, že je naše stránka správně vytvořena a validní dle HTML5.

Významným rozdílem, oproti předchozí verzi HTML, je **vypuštění nepotřebného a zjednodušení** v mnoha směrech. To má **příznivý dopad na přehlednost kódu, jeho jednoduchost a odstranění problémů se zápisem a kompatibilitou**.

Tímto však novinky tohoto standardu nekončí. Dále v této práci uvedu nové elementy, které standard přináší. Některé z nich také využívám v praktické části této práce. [15]

## 2.2 Elementy

Elementem se v kontextu HTML a v této práci myslí prvek, jenž je stavebním kamenem každé HTML stránky. Uvedu zde nejvýraznější novinky, včetně jejich zápisu a praktického použití.

Jednou z hlavních novinek je nativní podpora multimédií tedy audio a video souborů. V novém standardu již není třeba instalovat různé pluginy nebo rozšíření, abychom si mohli pustit oblíbenou píseň nebo videonahrávku. Vše lze již přehrávat pohodlně z prohlížeče bez zbytečných instalací.

Dále bych se rád zaměřil na element canvas, jakožto novinku v novém standardu.

### 2.2.1 Canvas

Tento element je základním elementem aplikace v praktické části, proto se na něj zaměřím trochu podrobněji.

Element Canvas, neboli česky plátno, je novým elementem HTML5. Používá se pro kreslení grafiky za běhu pomocí skriptování (obvykle JavaScriptem). Nemá totiž žádné kreslicí schopnosti, je to jen tzv. kontejner pro grafiku, která bude vykreslena nebo načtena pomocí JS.

---

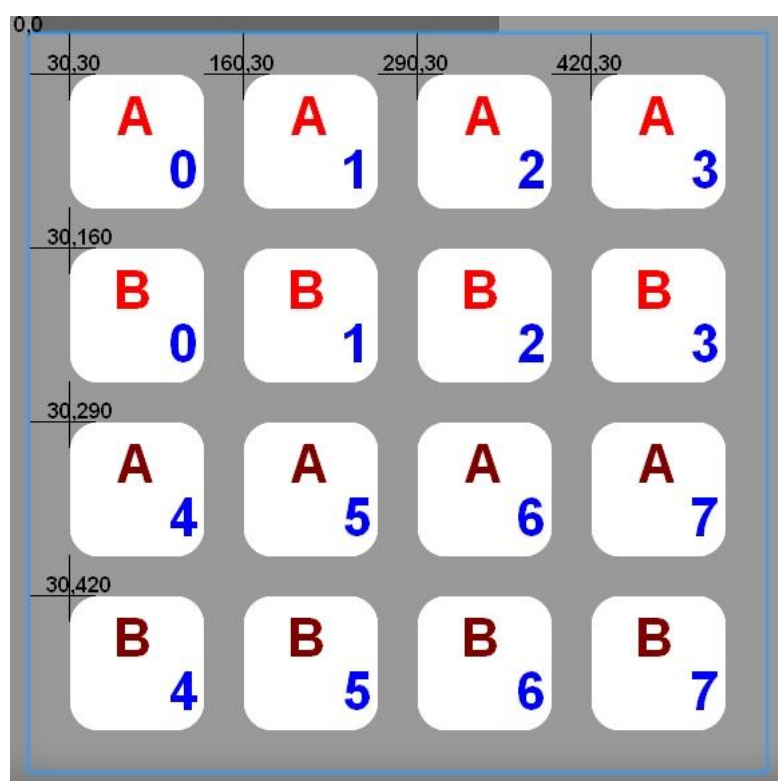
<sup>15</sup> <http://validator.w3.org/>

Princip plátna původně představila společnost Apple, za účelem vytvoření widgetů pracovní plochy pro Mac OS X. Problém tedy byl i s vlastnickými právy této společnosti. Nakonec však Apple souhlasil se zařazením do volné licence konsorcia W3C. [15]

Před příchodem elementu canvas bylo kreslení na webových stránkách, řešeno prostřednictvím pluginů a zásuvných modulů třetích stran. Např. pomocí Adobe Flash, SVG, VML a jiné. Byl tedy docela veliký problém, nakreslit na webové stránce i jednoduchou čáru.

### ***Vlastnosti***

Pokud se na webové stránce použije element canvas, vytvoří se obdélníková oblast. Defaultní neboli **výchozí velikost je na výšku 150 pixelů a na šířku 300 pixelů.** [15]



*Obrázek 9 – Souřadnice plátna  
(z praktické části této práce)*

Lze ale zadat velikosti vlastní, dle potřeby. A to buď přímo do vlastností elementu, nebo pomocí kaskádových stylů. Doporučuje se však zapisovat velikost canvasu přímo do atributů neboli vlastností. Důvodem je, že pokud bychom nezadali výšku a šířku přímo ve vlastnostech zápisu tagu elementu canvas v HTML, je výchozí

hodnotou šířky a výšky, jak bylo napsáno výše - 300px x 150px. V kaskádových stylech bychom potom nedefinovali reálnou výšku a šířku, ale pouze měřítko. A to by mohlo způsobovat problémy v zobrazení (rozptýlení obrazu na jednotlivé pixely). Výchozí hodnotou je také průhlednost.

Je-li element vložen do stránky, můžeme s ním pracovat. Důležitou součástí tohoto elementu je JS, který nám pomáhá do něj vykreslovat a přidávat různou grafiku. Můžeme v něm malovat, nebo přidávat a tvořit velice pokročilé animace.

Jedná se o průlomový prvek v tvorbě dvourozměrné (později možná i vícerozměrné) grafiky na webových stránkách. Spolu s JS se jedná o mocný vykreslovací nástroj, který někde i nahrazuje klasické použití obrázků a kaskádových stylů.

Abychom mohli plátno programově použít, musíme získat jeho tzv. kontext. Můžeme poté provádět různé akce a nakonec je na onen kontext aplikovat.

| Canvas-kontext                                                                                                                                                                                                                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>cnvs = document.getElementById('canvas').getContext('2d');<br/><br/>// Nalezení &lt;canvas&gt; elementu v DOM a následné načtení kontextu do proměnné<br/>cnvs (volání metody canvasu - getContext() s předáním řetězce „2d“)</pre> |

Souřadnice plátna (x, y), jak si můžeme všimnout na obrázku 8, začínají na hodnotách 0 – horizontálně, 0 – vertikálně v levém horním rohu, což je důležité si uvědomit pro následné vykreslování nebo tvorbu grafiky. [15]

I když se jedná o velmi mocný nástroj, není vhodné jej však používat v případech, kdy nám postačí jiný dostupný element. Například co se týče nadpisů nebo textu. Na to máme vyhrazené elementy jiné, které jsou určeny pro ten daný účel a použití složitě nastavovaného elementu canvas by bylo tudíž zbytečné.

Alternativou k elementu canvas je SVG, které slouží také k vykreslování složitější grafiky, ale je zaměřeno na grafiku vektorovou.

### **Použití:**

V praxi je element canvas na HTML stránce použit následujícím způsobem: V dokumentu je uvozen otevíracím tagem <canvas> a jelikož se jedná o tag párový, musí být také ukončen uzavíracím tagem s lomítkem </canvas>. Důležité je i zadání výšky a šířky ve vlastnostech. Pokud atributy ne zadáme, jsou brány výchozí hodnoty

elementu uvedené výše. K identifikování konkrétního canvasu je zadáno i ID, které je důležité i pro stylování kaskádovými styly. [11]

| Element canvas                                                                                                                                                                                              |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>&lt;canvas id="canvas" width="550" height="550"&gt;   &lt;span class="chyba"&gt;Váš prohlížeč zřejmě nepodporuje HTML5 - Canvas&lt;/span&gt; &lt;/canvas&gt; // standartní zápis elementu canvas</pre> |

Mezi otevíracím a uzavíracím tagem je vložena informace, která se zobrazí, pokud prohlížeč nerozumí zápisu, nebo neumí element canvas zobrazit.

### 2.2.2 Audio

Jak už název napovídá, stará se tento element o zprostředkování zvuku. HTML5 nám tedy umožňuje, standardní cestou bez použití jakéhokoliv plug-inu používat zvukové soubory na webových stránkách.

| Prohlížeč         | .ogg | .mp3 | .wav |
|-------------------|------|------|------|
| Chrome            | x    | x    | x    |
| Internet Explorer |      | x    |      |
| Firefox           | x    | x    | x    |
| Safari            |      | x    | x    |
| Opera             | x    | x    | x    |

Tabulka 1 – podpora formátů souborů elementu <audio> v jednotlivých prohlížečích [12]

#### Použití:

| Element audio                                                                                                                                                                                                                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>&lt;audio controls autoplay loop&gt; &lt;source src="hudba.mp3" type="audio/mp3"&gt; &lt;source src="hudba.wav" type="audio/wav"&gt; &lt;source src="hudba.ogg" type="audio/ogg"&gt; &lt;/audio&gt;  // Vložení všech 3 možných formátů souborů</pre> |

### 2.2.3 Video

Element video je dalším průkopníkem multimediálních elementů nového standardu. Konečně dostali tyto dva multimediální elementy (audio i video) konzistentní, standardizovanou podporu a zároveň nevyžadují zásuvný modul „plug-in“. Nepoužívá také žádnou ochranu proti kopírování.

| Formát videa   | MP4   | WebM   | Ogg    |
|----------------|-------|--------|--------|
| Kódování videa | H.264 | Vp8    | Theora |
| Kódování audia | AAC   | Vorbis | Vorbis |

Tabulka 2 – tři používané formáty a jejich kódování [11]

Stažení je stejně snadné jako stahování obrázků, jednoduchým pravým kliknutím na tlačítku myši.

Menším takovým problémem, však stále zůstává rozdílnost podporovaných formátů souborů, konkrétně tří. Typy souborů s používanými kodeky<sup>16</sup> můžete vidět v tabulce 2.

### **Použití:**

Jako každý element má i element video různé vlastnosti, které mu můžeme nastavit. V následujícím odstavci zdrojového kódu můžeme vidět samotné použití elementu video a jeho základních atributů.

| Element video                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>&lt;video //přidání ovládacích prvků pro řízení videa controls  //automatické přehrávání po načtení stránky Autoplay  //smyčka - po skončení videa automatické spuštění od začátku loop  //zdrojová složka s videem src="video/prvni.mp4"  //výška a šířka videa na stránce width="480" height="360"  //obrázek, který se zobrazí, pokud video není přehráváno poster="obrazky/uvodni.jpg"  //id pro video element, které můžeme použít pro stylování, nebo pozdější úpravu JS id="video"&gt; &lt;/video&gt;</pre> |

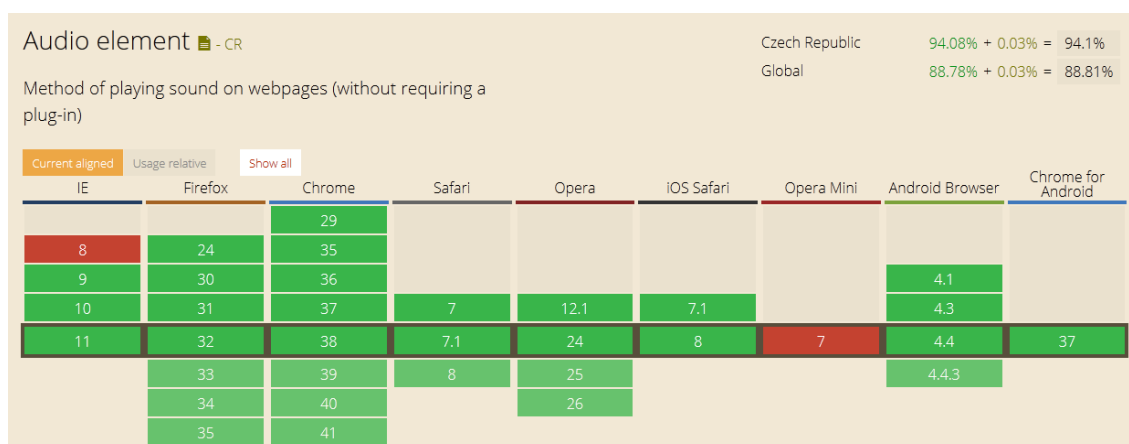
### **2.2.4 Podpora ze strany prohlížečů**

Dnes se používá okolo pěti hlavních majoritních prohlížečů od různých vývojářů. Myslím tím prohlížeče pro platformu osobních počítačů. Dále bychom mohli hovořit o prohlížečích pro mobilní přístroje, chytré telefony, tablety a jiná přenosná zařízení.

Ve většině moderních prohlížečů jsou základní elementy a jejich vlastnosti standardu HTML5 již v celku podporovány.

---

<sup>16</sup> Kodek – kompresní technologie o dvou složkách (kodéru a dekodéru). Slouží k redukování velikosti souboru, při zachování optimální kvality vjemu pro člověka (obrazu, zvuku, videa aj.)



Obrázek 10 – příklad podpory elementu Audio (10/2014)  
(Dostupné z: <http://caniuse.com/>)

Ovšem pokud chceme vytvořit aplikaci, u které bychom byli rádi, aby ji uživatelé spustili alespoň v nejvíce používaných prohlížečích, je třeba ověřit, zdali jsou všechny součásti podporovány ze strany prohlížeče.

Slouží nám k tomu, jako jeden z mnoha, výborný online nástroj<sup>17</sup>. Pomocí tohoto nástroje můžeme jednoduše zjistit, které elementy a funkce jsou pro jaký prohlížeč dostupné, popřípadě v jaké jeho verzi.

Příklad výsledku z tohoto nástroje můžeme vidět na Obrázek 10, kde máme zobrazenou podporu prohlížečů pro element Audio.

Zelené položky zobrazují podporováno, naopak červené nám říkají, že prohlížeč nekorektně nebo vůbec nespolupracuje s tímto elementem.

## 2.3 WAI-ARIA

Dalším poměrně novým standardem, se kterým HTML5 spolupracuje je tzv. specifikace WAI-ARIA (**Web Accessibility Initiative – Accessible Rich Internet Applications**), která získala status doporučení od konsorcia W3C – 20. března 2014. Jedná se o výborné vylepšení, které tvoří rozhraní mezi webovým prohlížečem a čtečkami obrazovek. Již nyní tedy můžeme tzv. „role“ přiřadit elementům v HTML stránce. [16]

Role je atribut, který přiřazujeme jednotlivým elementům. Popisuje strukturu webové stránky – její oblasti a také vlastnosti, v jakém stavu (rozbaleno, sbaleno) se jednotlivé widgety nachází (např. „menu“, „treeitem“ nebo „slider“).

<sup>17</sup> <http://caniuse.com/>

### Příklad:

| WAI-ARIA v HTML                                                   |
|-------------------------------------------------------------------|
| <code>&lt;li role="menuItem"&gt; Položka menu..&lt;/li&gt;</code> |

Tento standard tedy pomáhá a usnadňuje, osobám se zdravotním znevýhodněním, prostřednictvím nainstalované čtečky obrazovky, prohlížení webových stránek a hlavně orientaci na nich.

Jediným zádrhelem je, že není zatím podporována ve všech prohlížečích na 100%. Hlavně u starších verzí. Do budoucna by se však měla podpora zlepšovat, vzhledem k úplnému dokončení specifikace HTML5 a to 28. října 2014.

### 3 NÁSTROJE A SOFTWARE PRO TVORBU WEBOVÉ APLIKACE

V následujících řádcích se zaměřím na software pro úpravu grafiky, zvuku, prog. kódu a ladících nástrojů, které jsou nezbytné pro tvorbu moderní webové aplikace.

Základní věcí, kterou uživatel pro tvorbu jakékoliv aplikace potřebuje, je hlavně alespoň základní znalost onoho daného programovacího jazyka, který chce pro tvorbu aplikace použít, nebo který si vybere. V první kapitole jsem zmínil některé základní programovací jazyky, které se pro tvorbu webových aplikací dají využít. Ke každému programovacímu jazyku náleží určitá syntaxe, což jsou formální pravidla formulování příkazů počítačového jazyka (gramatická struktura), kterou by měl uživatel (programátor) znát.

Nezbytnou pomůckou je, alespoň pro začátek, mít po ruce příručku, či manuál, kde můžeme vyhledat potřebné informace a to buď v aktuální papírové, nebo elektronické podobě. Elektronická příručka je z hlediska aktuálnosti asi nejlepší, pokud se samozřejmě jedná o příručku, nebo manuál, o který se stará buď korporace, která jazyk vymyslela a provádí jeho úpravy a vylepšení, nebo komunita uživatelů, kteří do takového manuálu přispívají.

Důležité je také zjištění kompatibility se zařízeními, na kterých chceme vytvořenou aplikaci provozovat. Jestli zvolený jazyk, nebo způsob, kterým chceme aplikaci publikovat, je správný pro bezproblémový chod aplikace na cíleném zařízení.

Pro tvorbu webové aplikace je důležitá znalost webové tvorby. Jak používat elementy HTML, ke strukturování obsahu do nadpisů, seznamů, odstavců apod. Určitě je nutná i znalost kaskádových stylů, bez kterých se moderní web a webová aplikace jistě neobejde. Nakonec nejdůležitějším předpokladem pro tvorbu aplikace je znalost programovacího jazyka JavaScript. K samotné tvorbě HTML5 stránky není potřeba, ale pokud chceme využít pokročilejších funkcí elementů, jako je např. vykreslování grafiky na elementu canvas, nebo komunikace s webovým serverem, tak se bez něj jistě neobejdeme. O všech je napsáno mnoho literatury, speciálně věnované každému z nich s praktickými příklady. Některé zdroje můžete najít v seznamu použité literatury na konci této práce.

V následujících řádcích se budu zabývat základním a nejznámějším aplikačním softwarem pro práci s grafikou, který je pro tvorbu webových, ale i jiných aplikací nezbytný. Dále určitě softwarem pro tvorbu a editaci zdrojových kódů, pro práci se zvukem a nakonec nejpoužívanějšími webovými prohlížeči.

### 3.1 Grafické aplikace

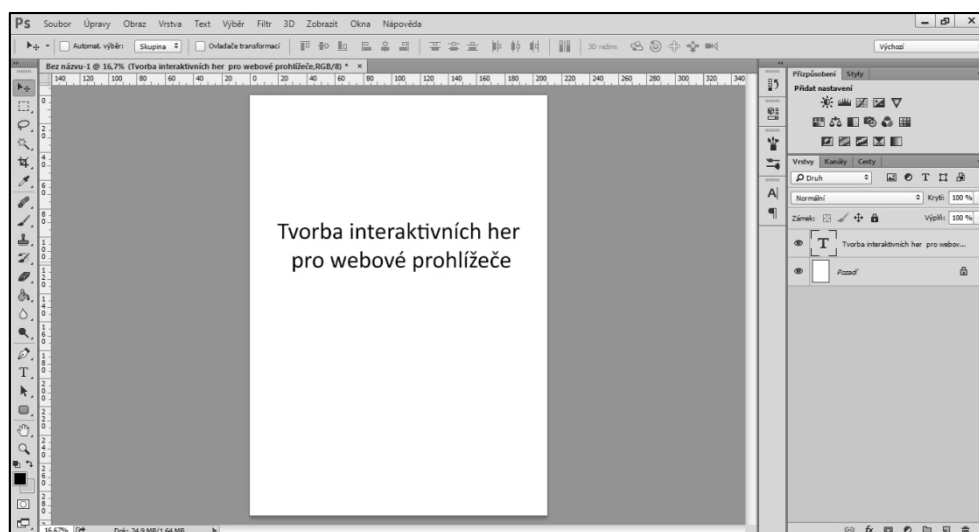
Slouží k úpravám a manipulaci s fotografiemi a jiným grafickým materiálem. Grafických programů je celá řada a mohli bychom je rozdělit dle různých hledisek. Dle zpracovávané grafiky – vektorové vs. rastrové, dle ceny – volně šiřitelné vs. komerční aplikace apod.

Zde zmíním nejpoužívanější aplikace pro úpravu grafiky, které se používají jak při tvorbě webu, tak v aplikacích na něm provozovaných.

Většinou se jedná o rastrovou grafiku, která je na webových stránkách používána, proto se budou následující řádky týkat pouze programů primárně používaných pro práci s rastrovou grafikou.

Při tvorbě jakékoliv webové stránky, potřebuje autor nějaký grafický program, ve kterém potřebuje vytvořit layout neboli grafický vzhled stránky.

Po vytvoření layoutu je skoro na každém webu použita galerie fotografií a obrázků, které je potřeba upravit. Obrázky a fotografie se musí připravit na takový druh publikování, z důvodu zmenšení velikosti a objemu přenášených dat skrze síť. K tomu samozřejmě opět potřebujeme nějaký program na úpravu grafiky.



Obrázek 11 - Prostředí programu Adobe Photoshop CC (Zdroj: snímek obrazovky)

### 3.1.1 Adobe Photoshop

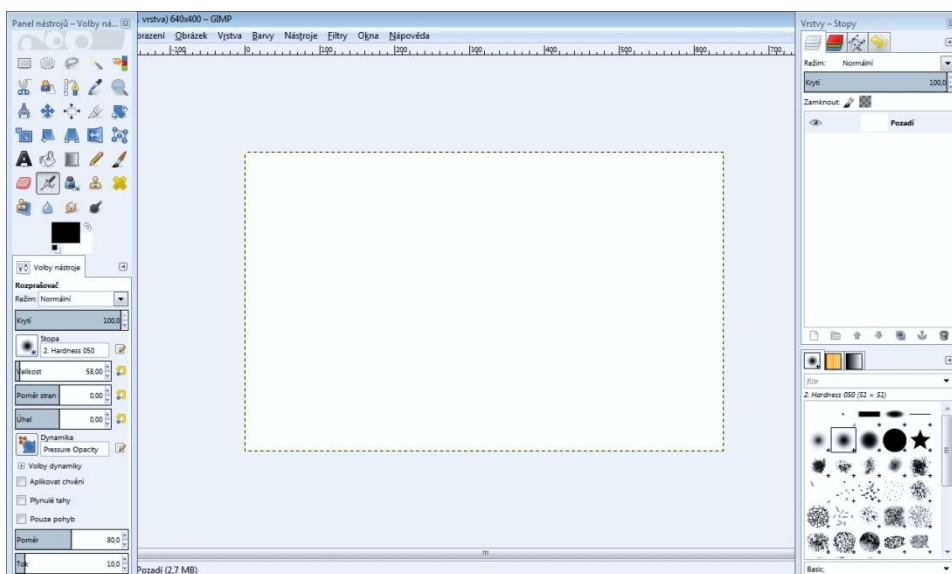
Software na tvorbu a editaci obrázků a fotografií vyvinutý společností Adobe. Je považován za jednoho z nejlepších editorů fotografií vůbec. Umožňuje uživatelům manipulovat, ořezávat, měnit velikost, barvu a mnoho dalších pokročilých úprav fotografií a obrázků. Je velice populární mezi fotografy i grafiky. Na obrázku 11 je prostředí verze CC (Creative Cloud).

Spustit jej můžeme jak na počítačích pod systémy Windows, tak i na počítačích firmy Apple s operačním systémem Mac OS. Pokud uživateli nějaká funkce chybí, je možné jej rozšířit o celou řadu pluginů neboli zásuvných modulů.

### 3.1.2 GIMP

GIMP je anglická zkratka - GNU Image Manipulation Program. Jedná se o volně distribuovaný software na úpravu, tvorbu a jinou manipulaci s fotografiemi a dalším grafickým materiálem. Ukrývá v sobě mnoho funkcí včetně rozšiřitelnosti o pluginy a proto má široké možnosti použití. [17]

GIMP je napsán a rozvíjen pod X11<sup>18</sup> na platformách UNIX. Ale v podstatě stejný kód běží také na platformě MS Windows a Mac OS X pod kterými se dá také používat. Jedná se o moderní multiplatformní software, navíc volně šiřitelný, což z něj dělá velice kvalitní a oblíbený nástroj pro práci s grafikou. Na obrázku 12 je jeho pracovní prostředí.



Obrázek 12 - pracovní prostředí programu GIMP 2.8 (Zdroj: snímek obrazovky)

<sup>18</sup> X11 – jedná se o systém oken. Běžný na Unixových OS

## 3.2 Editory zdrojových kódů

Editory webového obsahu můžeme rozdělit na tzv. strukturní a Wysiwyg. Strukturní editory slouží k přímému psaní a úpravě HTML kódu, PHP, JavaScriptu a jiných jazyků a kódů. Je tudíž nutné znát syntax zvoleného jazyka (např. tagy HTML). Jedná se většinou o prosté textové editory s některými pokročilými funkcemi barevného zvýraznění určitých částí textu (např. tagy, závorky aj.). Nejznámějším českým strukturním, volně šiřitelným editorem je editor PSPad Editor od Jana Fialy, nebo jednoduchý Notepad, který je součástí programového vybavení operačního systému Windows.

Wysiwyg<sup>19</sup> editory zobrazují v reálném čase, jak bude stránka vypadat a mají pro úpravu obsahu i své vlastní nástroje, podobně jako je tomu u textových editorů (např. MS Word). Autor tedy nemusí znát HTML jazyk, jeho kód je generován automaticky. U moderních editorů je však i možnost zásahu do samotného kódu a jeho úpravě.

Nejpoužívanějším a nejznámějším wysiwyg editorem je editor od společnosti Adobe – Dreamweaver. Dalším často používaným je Microsoft Expression Web. Dříve pod názvem Microsoft Frontpage.

### 3.2.1 PSPad

PSPad editor patří do skupiny tzv. freeware neboli volně šiřitelného softwaru. Stále však s uzavřeným kódem. Vyvíjí jej stále a od začátku (od roku 2001) pan Jan Fiala v programovacím jazyku Delphi.

Jedná se o textový univerzální editor pro MS Windows s mnoha funkcemi. Např. zvýrazňování syntaxe, porovnávání více souborů s výrazněním rozdílů, HEXA editor, podpora FTP a další. Podporuje mnoho jazyků (Pascal, C++, HTML, PHP, Python, SQL a další).

### 3.2.2 Adobe Dreamweaver

Standard ve vizuálním vytváření obsahu. Je součástí balíku produktů společnosti Adobe tzv. Adobe Creative Suite, jehož součástí je i Flash, Photoshop, Illustrator, InDesign, Acrobat a další.

---

<sup>19</sup> Wysiwyg – z angličtiny „What you see is what you get“ – označují se touto zkratkou editory u kterých je při tvorbě vidět, v reálném čase, výsledný dokument v jeho aktuální podobě.

Umožňuje hlavně začátečníkům začít s tvorbou webového obsahu, jelikož jim poskytuje mnoho užitečných nástrojů, které se snadno ovládají. Dreamweaver obsahuje kódovací nástroje a prvky pro HTML, XHTML, XML, CSS, JavaScript a umožňuje i upravovat zdrojový kód přímo.

### 3.3 Editace zvuku

Trojici používaných typů softwaru, pro tvorbu nejen webových aplikací, uzavírají aplikace pro tvorbu a úpravu zvuku. Ve všech aplikacích by měl být zvuk velice důležitým prvkem. Většina aplikací je, nebo by měla být, multimediálních. Tedy využít všechny možnosti k co nejlepšímu uživatelskému dojmu a tím i popularitě aplikace.

Určitě se v každé aplikaci najde nějaká příležitost k integraci zvuku. Ať už jako nějaké upozornění, oznámení, nebo odpověď na nějakou akci uživatele.

Určitě je důležité zvážit, jestli je vložení zvuku skutečně nutné, aby to uživatele zbytečně nerušilo a nebylo hlavním důvodem, proč aplikaci nechce používat.

Na mnoha webových stránkách je vložena hudba v pozadí, se záměrem webovou stránku určitým způsobem vylepšit. Uživatele v mnoha případech ruší při vyhledávání informací a není-li možnost hudbu na stránce vypnout, může se web stát nezajímavým a v případě firemního webu se může jednat i o ztrátu potencionálních zákazníků.

#### 3.3.1 Audacity

Audacity je volně šiřitelný, jednoduše obsluhovatelný, multi-trackový audio editor a rekordér pro OS Windows, Mac OS X, GNU/Linux a další operační systémy.

Patří k nejlepšímu softwaru na jednoduchou úpravu, nahrávání a editaci zvuku. Je šířen pod licencí GNU<sup>20</sup>. Na obrázku 13 je možné vidět pracovní prostředí programu.

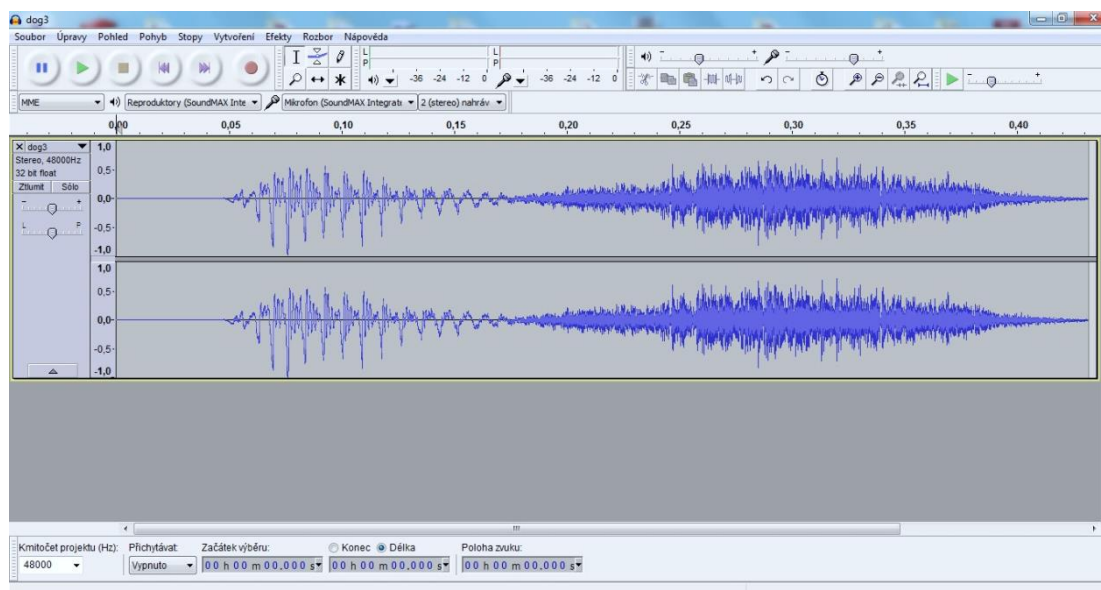
Umožňuje:

- Nahrávání zvuku
- Převést magnetofonové pásky nebo kazety do digitální formátu
- Editovat zvukové soubory WAV, AIFF, FLAC, MP2, MP3 nebo Ogg Vorbis

---

<sup>20</sup> GNU General Public License – licence pro svobodný software. Volně dostupné zdrojové kódy. Možnost vlastní úpravy softwaru

- Aplikovat na nahrávku mnoho integrovaných efektů a mnoho dalšího [18]



Obrázek 13 – Pracovní prostředí programu Audacity (Zdroj: snímek obrazovky)

### 3.4 Webový prohlížeč

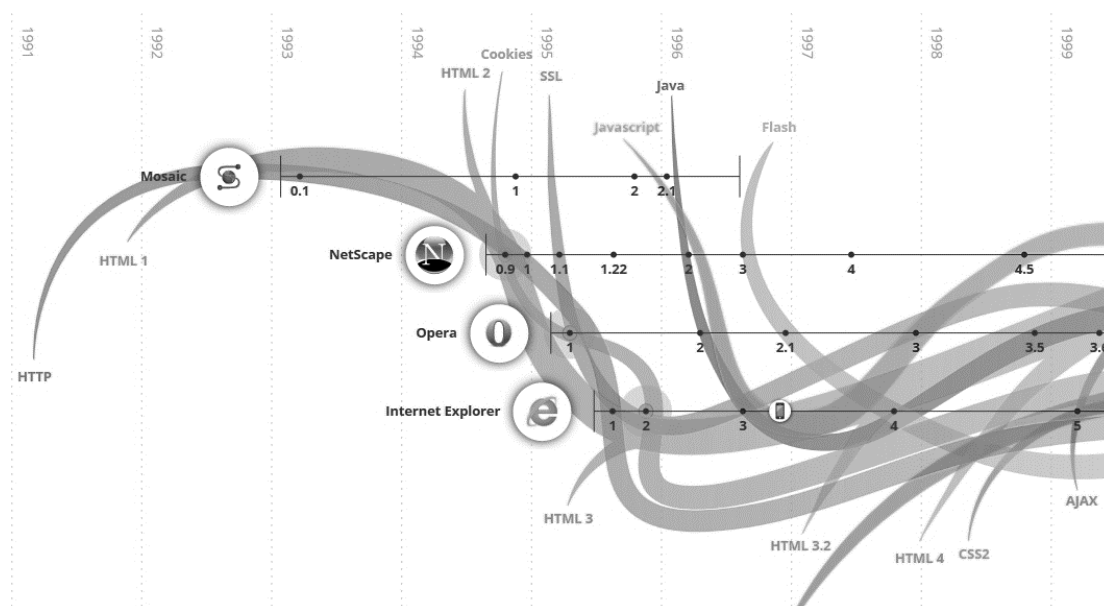
Prohlížečů webových stránek existuje ohromné množství a každý si může vybrat, který mu vyhovuje. Pro prohlížení webových stránek a spouštění webových aplikací je to velice důležitý software, který by neměl na žádném počítači připojeném k internetu určitě chybět.

Od doby, kdy se začaly používat client-side skriptovací jazyky, se z webového prohlížeče stává i velmi schopný interpreter. Zejména jazyka JavaScript, ze kterého se po uvedení nových elementů HTML5, stal hlavní ovládací jazyk jejich funkcí.

Skrze webový prohlížeč se dostáváme k všemožným službám, které nám internet nabízí (hlavně www a e-mail).

Mezi nejpoužívanější prohlížeče na platformě PC můžeme zařadit:

- Internet Explorer
- Mozilla Firefox
- Google Chrome
- Opera
- Safari



Obrázek 14 – Časová osa prvních webových prohlížečů  
(Dostupné z <http://evolutionofweb.appspot.com/>)

Historie webových prohlížečů se začala psát roku 1992, kdy na svět přišel Mosaic. Dále pak Netscape (předchůdce dnešní Mozilly Firefox) a v roce 1995 Internet Explorer. Společnost Apple vydala svůj první prohlížeč Safari v roce 2003 a později v roce 2008 spatřil světlo světa první Google Chrome. Na obrázku 14 je časová osa prvních webových prohlížečů.

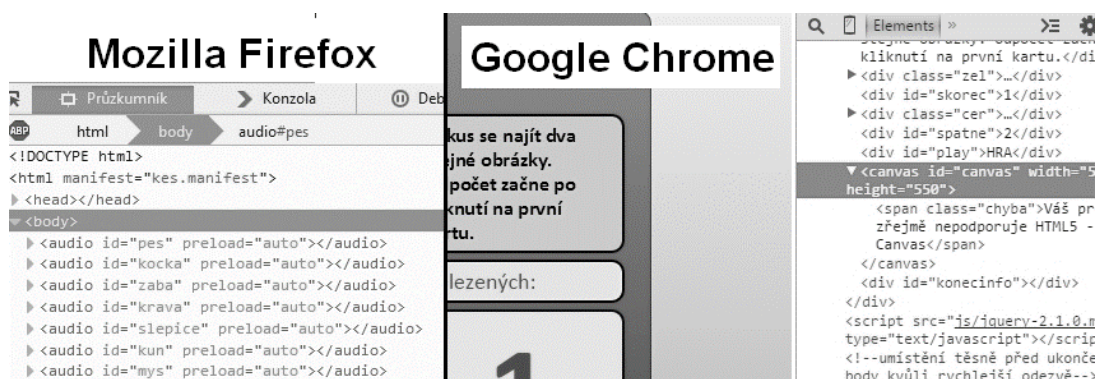
Nejvíce potíží způsobuje tvůrcům internetových stránek a aplikací prohlížeč Internet Explorer, který hlavně v minulosti nedodržoval téměř žádný z webových standardů, a všechny stránky (pokud chtěl autor zprostředkovat obsah co nejširšímu publiku) musely být dle něj optimalizovány. To způsobovalo nemalé problémy se zobrazováním stránek v ostatních prohlížečích. Dnes se již situace zlepšila, ale stále je Internet Explorer pod neustálou kritikou, i z hlediska bezpečnostních děr, které obsahuje.

Osobně tento prohlížeč nerad používám a volím raději některou z vypsanych alternativ z výše zmíněných důvodů.

Dle globálních statistik StatCounter (<http://gs.statcounter.com/>) k březnu 2015 je nejpoužívanějším a tedy i nejpopulárnějším prohlížečem Google Chrome s podílem 48.95%, daleko za ním je Internet Explorer s 17.91%, dále Mozilla Firefox 16.83%, Safari 10.96% a na konci je Opera 1.63% a ostatní prohlížeče.

### 3.4.1 Vývojářské nástroje

Velice důležitým doplňkem, dnes již každého prohlížeče, je vývojářský nástroj neboli v angličtině web development tool.



Obrázek 15 – Vývojářské nástroje prohlížečů Mozilla Firefox (vlevo) a Google Chrome (vpravo) (Zdroj: snímek obrazovky)

Každý prohlížeč obsahuje vlastní integrovaný vývojářský nástroj, nebo je možnost si jej doinstalovat jako doplněk. Nejznámějším propracovaným doplňkem pro Mozillu Firefox je Firebug.

Jak vypadají integrované v prohlížeči, můžete vidět na obrázku 15. Většinou se zapínají klávesou F12, nebo je možnost jejich vyvolání skrze menu.

Jedná se o důležitý doplněk, kterým mohou vývojáři webových aplikací sledovat, testovat a ladit své aplikace. Pracují s CSS, HTML, DOM strukturou, JavaScriptem, obrázky a dalšími komponenty se kterým pracuje webový prohlížeč.

JavaScriptový kód se dá pomocí tzv. breakpointů<sup>21</sup> pozastavit a můžeme například vidět, jakých hodnot nabývá ta která proměnná, nebo jak se chová naprogramovaný cyklus, což pomáhá k vyladění a odstranění chyb. Této funkci nebo nástroji se říká debugger<sup>22</sup>.

Kaskádové styly se ve vývojářském nástroji dají v reálném čase upravovat, takže vývojář hned vidí, co který styl, nebo vlastnost s daným prvkem dělá, nebo naopak proč nefunguje.

<sup>21</sup> Breakpoint – bod přerušení. Místo, kde dojde k pozastavení kódu

<sup>22</sup> Debugger – nástroj pro hledání chyb

### 3.5 Typy her v HTML5

Jaké typy her lze vůbec v HTML5 vytvářet? V HTML5 a konkrétně jeho elementu canvas, lze vytvořit dnes již prakticky cokoli. Dá se již pracovat nejen s druhým rozměrem (2D), ale můžeme vytvářet hry i v rozměru třetím (3D) – tzv. WebGL.

Díky technologii WebSockets se dají vytvářet i hry pro více hráčů pro hru v reálném čase skrze internet.

Rád bych stručně upozornil na několik typů her, na kterých lze vidět, jak daleko se dá v tvorbě webové aplikace (hry) zajít.

#### ***Bejeweled***

Ke konci roku 2011, PopCap vydala HTML5 verzi své populární logické hry Bejeweled (obr. 16). Využívá technologii WebGL, zajišťující akcelerovanou grafiku.

K vyzkoušení: <http://bejeweled.popcap.com/html5>



Obrázek 16 – hra Bejeweled (Zdroj: snímek obrazovky)

#### ***WebQuake***

Další velice známá hra (obr. 17), předělaná skupinou SiPlus do prostředí HTML5. Využívá také technologii WebGL. Informace o kompatibilitě s prohlížeči, instalaci apod. na stránce projektu<sup>23</sup>.

K vyzkoušení: <http://webquake.quaddicted.com/Client/WebQuake.htm>

---

<sup>23</sup> <https://github.com/SiPlus/WebQuake>

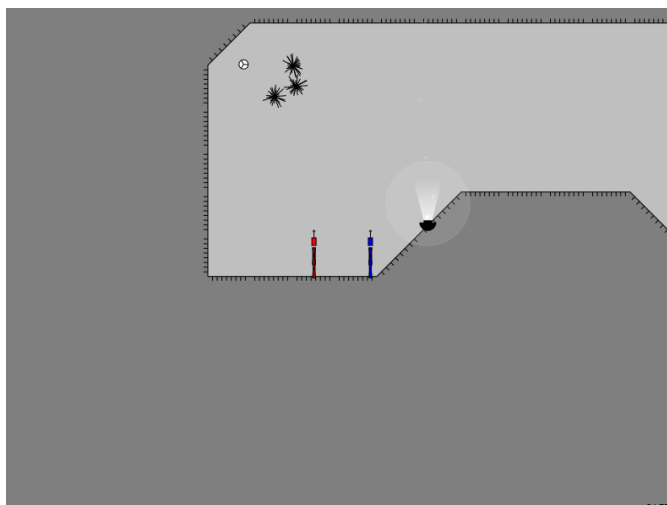


Obrázek 17 – Quake v elementu canvas HTML5 (Zdroj: snímek obrazovky)

### ***Robots Are People Too***

Posledním příkladem je multiplayerová hra s názvem Robots Are People Too. Hra je určena dvěma hráčům, kteří spolu musí spolupracovat, aby dosáhli stanoveného cíle. Hra je zvláštní v rozdělení obrazovky pokud se jeden hráč nachází jinde než ten druhý. Využívá také element canvas, neboli plátno.

K vyzkoušení: <http://raptis.com>



Obrázek 18 – Robots Are People Too (Zdroj: snímek obrazovky)

### 3.6 Omezení HTML5

I v standardu HTML5 existuje určité omezení. Kromě těch, kterými se budu dále zabývat, je omezením hardware, na kterém je prohlížeč a následně obsah webu spouštěn. Např. vykreslování grafiky, je závislé na vybavení grafickou kartou daného zařízení. Čím složitější grafika, tím větší zátěž pro procesor a GPU. [19]

Omezení jsou jinak individuální. Dána používaným zařízením a instalovaným OS.

#### ***Chybějící funkce***

Jedním z problémů HTML5 je, že množství funkcí poskytovaných zařízením (většinou těch mobilních), přístupných skrze HTML5, je limitováno. Tento limit lze pro aplikace, které vyžadují přístup k funkcím zařízení (mimo standard HTML5) přemostit pomocí pluginů neboli zásuvných modulů. Stejně jako u desktopových prohlížečů, pluginy pro mobilní přístroje jsou menší kus kódu, který lze snadno přenést v různých prohlížečích a platformách podle potřeby, a lze k nim přistupovat pomocí jazyka JavaScript přímo z aplikace HTML5. Pluginy jako je např. PhoneGap úspěšně překlenou mezeru konkrétní funkce zařízení a za pomoci komunitní podpory projektu, která je udržována skrze všechny hlavní OS a platformy prohlížečů, je snadno rozšiřitelná podle potřeby pro konkrétní aplikaci. [19]

#### ***Bezpečnost***

Základním problémem se spouštěním a interpretováním JS aplikací na straně klienta (client-side) je, že v konečném důsledku, má uživatel kontrolu nad běžícím kódem na jeho přístroji. V případě webových aplikací, přichází sám prohlížeč s výbornými výše zmíněnými ladícími nástroji a jejich ovládání je velice jednoduché.

S debuggerem JavaScriptu, jako je např. Firebug pro prohlížeč Mozilla Firefox, může kdokoli, koho to zajímá, studovat jak Facebook, Google nebo jiné webové stránky fungují. Je to výborné pro ladění a učení, ale jedná se o veliký problém, co se týče bezpečnosti.

Kdokoli pak může změnit, podstrčit jakoukoliv proměnnou v kódu. Např. informaci o geografickém umístění prohlížeče, nebo jiné daleko důležitější hodnoty.

Možnost upravovat takto webovou aplikaci, navíc přímo v prohlížeči, je o mnoho snadnější, než co se týče kódu nativního. [19]

### ***Element Audio***

Omezení audio elementu HTML5 je hlavně v tom, že jeho cílem je ve skutečnosti přehrávat jednotlivé zvukové soubory, jako např. hudbu na pozadí ve hře. Není vhodný pro zvukové efekty, zejména v případě, že jsou rychlé a pokud jich je mnoho k přehrání v jeden časový okamžik.

## II. Praktická část

---

### 4 WEBOVÁ APLIKACE PEXESO

Praktickým příkladem využití elementu canvas, nového standardu značkovacího jazyka HTML5, společně s JavaScriptem a jeho knihovnami, je má aplikace na téma známé karetní hry PEXESO. Na základě poznatků z teoretické části této práce, se v této kapitole pustím do tvorby vlastní webové aplikace, kde rozeberu hlavní části a princip jejího fungování.

#### 4.1 Historie a pravidla hry PEXESO

Klasické PEXESO netřeba dlouze představovat. Jedná se o známou karetní hru, v níž se trénuje hlavně paměť a soustředění. Hraje se většinou s 32 páry karet, tedy 64 kartami. Vymyslel ji v roce 1965 pan Zdeněk Princ. [19]

Nejmenší počet hráčů, jež se mohou hry zúčastnit, jsou dva. Český název této karetní hry vznikl ze sousloví - PEKelně SE SOustředě.

Pravidla jsou jednoduchá. Hra spočívá v nalezení stejných dvojic obrázků. Na začátku se všechny karty otočí rubem nahoru a zamíchají tak, aby nikdo z hráčů nevěděl, kde se která karta nachází. Hráč, který je na řadě, postupně otáčí dvě karty, které si vybere, tak aby na ně ostatní hráči viděli a mohli si zapamatovat jejich pozici. Pokud se karty shodují, sejmou se z hracího pole a hráč otáčí další dvojici. Pokud se karty neshodují, karty se otáčí zpět rubem vzhůru a pokračuje další hráč na řadě. Takto se pokračuje do té doby, než jsou všechny karty sejmuty z hracího pole.

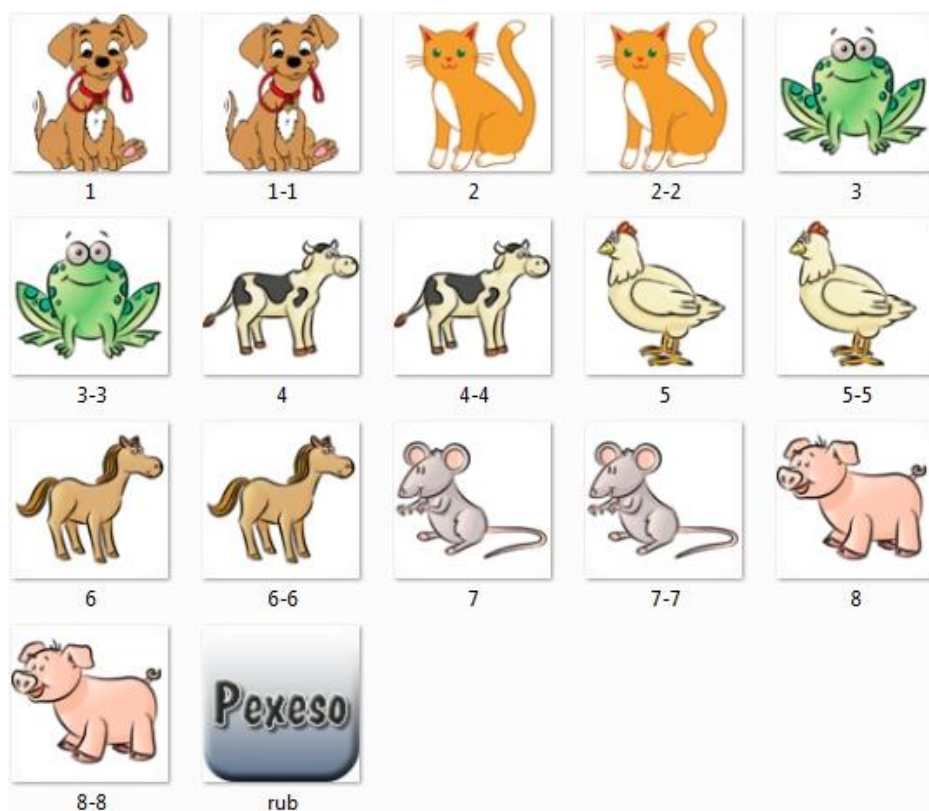
Existuje mnoho verzí a vylepšení např. pro nevidomé a neslyšící.

#### 4.2 Interaktivní hra PEXESO

Má praktická část práce je vytvořena dle pravidel a s použitím elementů standardu jazyka HTML5, kaskádových stylů, programovacího jazyka JavaScript a jeho známé knihovny jQuery.

Pracuji celkem s 8 páry obrázků a jednoho, který je použit na rub karet. Obrázky jsou na vyobrazeny na obrázku 16. Aby byla hra multimediální a tudíž uživatelsky

přívětivá, je otáčení karet doprovázeno hlasem, který uživateli říká, co se na kartě nachází. Konkrétně jde o názvy jednotlivých druhů zvířat.



Obrázek 19 – Obrázky použité v pexesu včetně rubové strany (Zdroj: snímek obrazovky)

#### 4.2.1 Zvuk

Zvuk byl vygenerován za pomoci online nástroje<sup>24</sup>, hlasem Eliska, technikou text to speech, neboli převod textu na řeč, kdy po napsání jakéhokoliv textu, jej tato aplikace přeloží na zvuk. Vloženy jsou i jiné dva zvuky a to po úspěšném nalezení všech dvojic, nebo naopak po vypršení časového limitu<sup>25</sup>. Možností vylepšení, co se týká zvuku, by jistě bylo přidání zvuku po nalezení stejné dvojice, nebo naopak po nenalezení stejných obrázků např. výborně, zkus to znova a podobně. Muselo by se však počítat s časovou prodlevou, aby zvuk nebyl přehráván v době, kdy je druhá karta otáčena a přehráván zvuk vyobrazeného zvířete. Určitě by bylo vhodné, vložit několik verzí oznámení úspěchu nebo neúspěchu, aby uživatele nerušil stále stejný zvuk.

<sup>24</sup> <http://www.acapela-group.com/>

<sup>25</sup> <http://www.pond5.com>

```

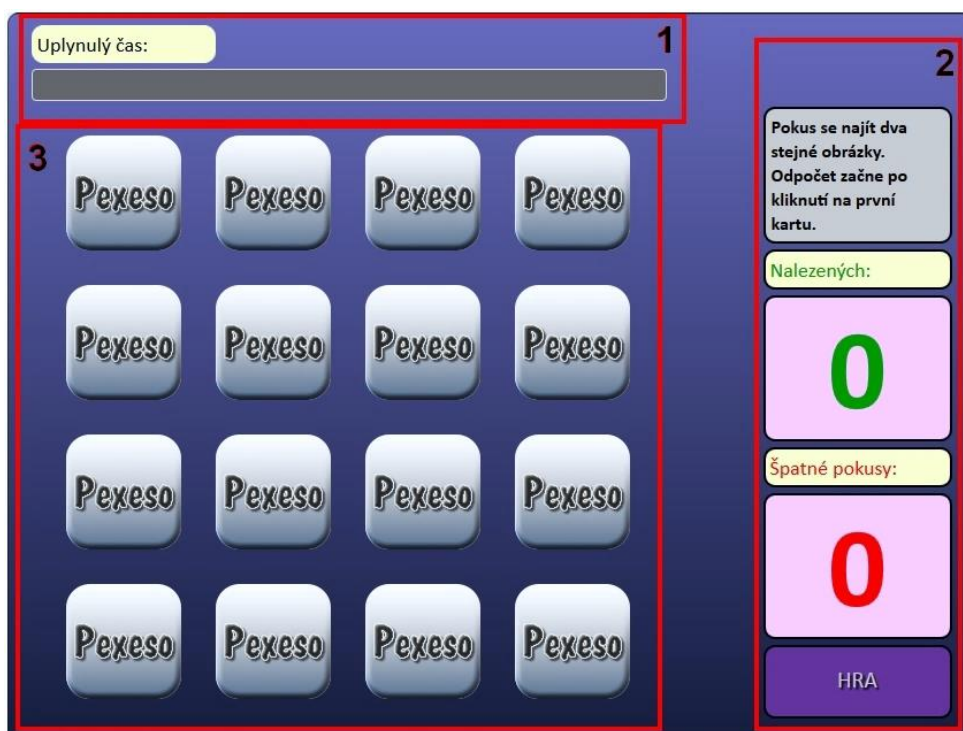
index.html
<audio id="prase" preload="auto">
  <source src="audio/wav/prase.wav" type="audio/wav">
  <source src="audio/mp3/prase3.mp3" type="audio/mpeg">
</audio>

```

Zápis elementu audio se dvěma audio formáty, ze kterých si prohlížeč vybere ten, který umí přehrát.

#### 4.2.2 Prostředí hry

Na obrázku 17 je zobrazeno prostředí hry ve webovém prohlížeči. Vyobrazeno je již po stisknutí tlačítka HRA, kdy je na elementu canvas vykresleno zamíchané hrací pole. Obrázek 17 je rozdělen na 3 sekce.



Obrázek 20 – Prostředí hry (Zdroj: snímek obrazovky)

V sekci 1 je umístěn časovač, který začne odpočítávat po kliknutí na první libovolnou kartu. Odpočítávání je řešeno za pomoci progressbar widgetu, knihovny jQuery, který zobrazuje procentuální část úplnosti nějakého procesu. Čas se zobrazuje zvětšováním oranžové výplně směrem doprava. Defaultně je čas nastaven na 61 vteřin. 60 vteřin je nastavení časovače, kdy se postup mění každých 600ms a o 1 vteřinu je nastaveno zpoždění volání funkce odpočtu od kliknutí na první kartu. Po uplynutí tohoto času je zavolána funkce **prohra()** a hráči je zobrazen neúspěch ve hře a jeho statistika. Část funkce, která má odpočet na starosti je ukázána níže s číslem řádku, kde

ji můžete nalézt ve zdrojovém kódu. Barva postupu se může změnit v kaskádových stylech (style.css) ve třídě ui-widget-header.

```
js/pexeso.js
304. function aktualpostup() {
305.     postup = $("#postupbar").progressbar("option","value");
306.     if (postup < 100) {
307.         $("#postupbar").progressbar("option", "value", postup + 1);
308.         odecetcas = setTimeout(aktualPostup, 600);
...

```

V sekci 2 je v horní části jednoduchý návod, jak hru hrát a jaké jsou pravidla. Níže se pak ukazuje počet nalezených dvojic a počet pokusů o jejich nalezení. Pokud jsou všechny dvojice nalezeny, je hráči oznámen úspěch ve hře a jeho statistika. Po kliknutí na tlačítko HRA jsou karty zamíchány a znovu vykresleny na elementu canvas a tím začíná hra od začátku.

V sekci 3 je umístěn element canvas, na kterém je všechno generováno a probíhá na něm samotná hra (otáčení obrázků, vyhodnocení hry).

Všechny vložené elementy na stránce, jsou vystylované pouze za pomoci CSS a jejich možností. Není použit žádný obrázek na jejich vzhled a není tedy zbytečně zpomalováno načítání samotné stránky.

JavaScriptový kód je dle zvyklosti umístěn na konci stránky, tedy těsně před ukončením elementu body. Zajištěna je tím rychlejší odezva, kdy se první načte celá stránka a poté se až začne vyhodnocovat naprogramovaný kód.

### 4.2.3 Programování hry

Nyní se tedy dostávám k samotnému programování hry. Uvedu zde jen některé pasáže kódu, které jsou důležité pro pochopení samotného fungování hry a její logiky.

Ve druhé kapitole jsem zmínil, jakým způsobem se vkládají externí soubory, jako jsou kaskádové styly a soubory jazyka JS, proto je zde dále neuvádím. Popsání syntaxe a sémantiky JS by dalo na další práci, proto vycházím alespoň ze základní znalosti a jednotlivé části nepopisuji do detailu.

Kód je rozdělen pro přehlednost do několika JS funkcí, které jsou postupně volány, dle jejich potřeby.

## HTML

Začínám s deklarací elementu canvas v samotné html stránce, kterou můžeme vytvořit za pomoci jakéhokoliv textového editoru, nebo nástrojů zmíněných v kapitole 3 této práce. Název stránky *index.html* je výchozím názvem. Jedná se o hlavní stránku a také o nejčastější stránku, kterou uživatel webu navštíví.

Úplně na začátku html stránky po deklarování doctype, je přímo v uvozovací značce `<html>` uveden atribut `manifest`, který specifikuje umístění souboru (*kes.manifest*), ve kterém je umístěn seznam s pravidly, pro uložení používaných souborů do cache webového prohlížeče.

```
index.html
<html manifest="kes.manifest">
```

Dále je zapsán element canvas s identifikátorem „canvas“. Šířka i výška elementu canvas je nastavena na 550px. Uvnitř párového tagu je zakomponována informace pro prohlížeče, které neznají element canvas a tudíž se v nich zobrazí, že jej nepodporují.

```
index.html
<canvas id="canvas" width="550" height="550">
  <span class="chyba">Váš prohlížeč zřejmě nepodporuje element HTML5 -
  Canvas</span>
</canvas>
```

## Modernizr

Pro demonstraci detekce podpory jednotlivých elementů ve webovém prohlížeči, je v kořenovém adresáři hry, přidán soubor „modernizr.html“, ve kterém jsou detekovány elementy uvedené v této práci (audio, video a canvas). Z důvodu velikosti externího souboru knihovny, kde nepotřebuji testovat všechno, co knihovna obsahuje, jsem si na webové stránce<sup>26</sup> vygeneroval vlastní část knihovny, která obsahuje pouze testování dostupnosti výše zmíněných elementů. Testování by se dalo umístit přímo do kódu aplikace, ale zvolil jsem pro jednoduchost externí metodu.

Níže příklad kódu pro testování podpory elementu canvas.

```
Modernizr.html
//v hlavičce souboru HTML
<script src="js/modernizr-moje.js"></script>

<p>Váš prohlížeč element canvas: <span id="canva"></span></p>
<script>
var canva = document.getElementById("canva"),
if (Modernizr.canvas) {
  canva.innerHTML = "Výborně podporuje element canvas";
```

<sup>26</sup> <http://modernizr.com/download/>

```
} else {  
    canva.innerHTML = "Váš prohlížeč nepodporuje element canvas";  
}  
</script>
```

### JavaScript

V souboru jazyka JS, jehož koncovka je .js, jsem začal deklarováním globálních proměnných, které jsou umístěny na začátku souboru.

```
js/pexeso.js  
1. var kontext, canvas;  
2. var prvnivyber = true;  
...  
7. var sirkakarty = 100;  
8. var vyskakarty = 100;  
...  
20. var pary = [ ["obr/1.png", "obr/1-1.png"],  
21.             ["obr/2.png", "obr/2-2.png"],  
22.             ["obr/3.png", "obr/3-3.png"],  
23.             ["obr/4.png", "obr/4-4.png"],  
24.             ["obr/5.png", "obr/5-5.png"],  
25.             ["obr/6.png", "obr/6-6.png"],  
26.             ["obr/7.png", "obr/7-7.png"],  
27.             ["obr/8.png", "obr/8-8.png"]];  
  
59. canvas = document.getElementById('canvas');  
60. kontext = canvas.getContext('2d');
```

Jedná se o proměnné používané v samotném kódu. Postupně se přidávají dle potřeby a využít se dají odkudkoliv v programu. Proměnnou může být číslo, text, pravdivostní hodnota, ale i pole hodnot, které se také nachází v programu pexesa.

Na řádce číslo 2, povoluji v proměnné **prvnivyber**, první výběr karty pravdivostní hodnotou true, neboli pravda. Dále si deklaruji další pomocné proměnné a na řádce 7 a 8, zadávám výšku a šířku samotných karet na plátně. Definování proměnných v programu, je zakončeno deklarovaným dvourozměrným polem, kde jsou uloženy po dvojicích cesty k obrázkům jednotlivých karet.

V řádce 59 dostávám do předem deklarované proměnné, element canvas s id=canvas a do proměnné kontext na řádce 60, umisťuji metodu **getContext ('2d')**, která vrací veškerou vykreslovanou 2d grafiku na elementu canvas. Existuje více kontextů a další budou jistě přibývat. Např. kontext „**webgl**“ rozšiřuje možnosti canvasu, o renderování grafiky ve třetím rozměru.

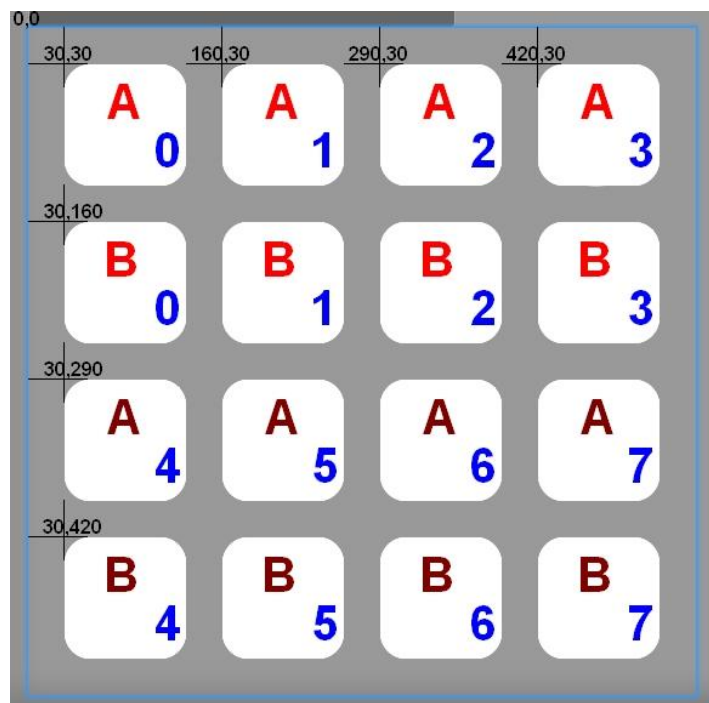
Další zajímavou částí kódu je konstruktor objektu s metodou.

```
js/pexeso.js
71. function Karta(sourX, sourY, kartasirka, kartavyska, obr, info) {
72.
73.     this.sourX = sourX;
74.     this.sourY = sourY;
75.     this.kartasirka = kartasirka;
76.     this.kartavyska = kartavyska;
77.     this.info = info;
78.     this.obr = obr;
79.     this.vykresli = zobrazrub;
80. }
81.
82. function zobrazrub() {
83.
84.     if (startovnicas === 1) {
85. kontext.clearRect(this.sourX,this.sourY,this.kartasirka,this.kartavyska);
86.     }
87.
88.     kontext.drawImage(rub,this.sourX,this.sourY,this.kartasirka,this.karta
vyska);
89. }
...
109. obrA = new Image();
110. obrA.src = pary[i][0];
111. kartaA = new Karta(mezX, mezY, sirkakarty, vyskakarty, obrA, i);
112. polekarty.push(kartaA);
...
121. kartaA.vykresli();
```

Konstruktor je funkce s parametry, která vytváří objekty. Zpravidla jeho název začíná velkým písmenem. V tomto kódu se vytváří konstruktor objektu **Karta**, kde se definuje způsob ukládání důležitých a potřebných informací, týkající se umístění, velikosti, adresy obrázku (řádky 71-78). Součástí objektu, je i metoda vykresli na řádku 79. Metodou se nazývá funkce v objektu.

Vykreslením karet na plátně předchází vytvoření elementu obrázku za pomoci JS na řádku 109. Cesta k souboru je vytažena z dvourozměrného pole **pary**, definovaného na začátku dokumentu. K vytvoření objektu **kartaA** dojde na řádku 111 s hodnotami, které jsou dopočítány tak, aby došlo k přesnému umístění karty na plátně (viz. obr. 18). Stejným způsobem je vytvořen i objekt **kartaB** (do páru).

Karty jsou postupně přidávány z levého horního rohu, kde první z páru je přidán a pod něj je přidán druhý do dvojice (dle označení  $A_0$ ,  $B_0$  atd.). Hodnoty jednotlivých karet jsou prozatím vloženy do pole s názvem **polekarty** a k samotnému vykreslení na



Obrázek 21 – Postupné vykreslení všech karet na plátně  
(Zdroj: autor)

plátně dojde, po volání metody vykresli na řádku 118. Nejde zatím o obrázkové karty, ale pouze o jejich rub.

V následujícím kódu dojde k zamíchání karet.

```
js/pexeso.js
123. function zamichat() {
124.
125.     var a, b, i;
126.     var mezinfr;
127.     var mezoobr;
128.     var velpole = polekarty.length;
129.
130.     for (i = 0; i < velpole * 3 ; i++){
131.
132.         a = Math.floor(Math.random()*velpole);
133.         b = Math.floor(Math.random()*velpole);
134.
135.         mezinfr = polekarty[a].info;
136.         mezoobr = polekarty[a].obr;
137.
138.         polekarty[a].info = polekarty[b].info;
139.         polekarty[a].obr = polekarty[b].obr;
```

```

140.
141.         polekarty[b].info = mezinfn;
142.         polekarty[b].obr = mezoobr;
143.     }
144. }

```

Po deklaraci funkce dochází k zápisu lokálních proměnných (ke kterým se dá přistoupit pouze uvnitř funkce). Do proměnné **vepole**, je uložena velikost pole **polekarty**, která má hodnotu 16 (dle celkového počtu hracích karet).

Cyklem for, který je nastaven na počet průchodů hodnotou 48 (3 x velikost pole **polekarty**), z důvodu důkladného zamíchání hracích karet.

Do hodnot **a** a **b**, které jsou součástí cyklu, je vloženo náhodně generované číslo (0-1) násobené hodnotou 16 (velikost pole) a zaokrouhlené na celá čísla směrem dolů.

Proměnné **mezinfn** a **mezoobr** fungují pro uložení mezihodnot **i** a cesty k obrázku jednotlivých karet, jejichž informace je uložena v poli **polekarty**.

Na řádku 138 se jednotlivé hodnoty přehodí. Hodnota **i** z pozice pole dle generovaného **a**, za hodnotu **i** z pozice pole **polekarty** dle generovaného čísla **b**.

Nakonec se na pozici (dle vygenerovaného čísla **b**), pole **polekarty**, za hodnotu **i** dosadí uložené mezihodnoty info a cesty k obrázku.

#### 4.2.4 Offline

Hra obsahuje i malý soubor „kes.manifest“, ve kterém jsou zapsány adresy souborů, které si má prohlížeč stáhnout do cache, aby byla hra funkční i po odpojení od sítě (pokud je hra spouštěna skrze internet).

Jedná se o všechny soubory, které jsou nezbytné pro bezproblémový chod hry. Níže pár řádků ze zmíněného souboru.

```

kes.manifest
CACHE MANIFEST
#verze 1.0

//# se označuje komentář
CACHE:
#ikona
ikona/favicon.ico

#stranky
index.html

#skripty & styly
css/style.css
js/pexeso.js

```

## 4.3 Komu je hra určena a její složitost

Hra je určena všem hráčům, kteří alespoň trochu znají a mají rádi karetní hru PEXESO. Obrázky byly zvoleny v tématu zvířat, navíc doprovázené zvukem, takže s nalezením stejné dvojice, by neměly mít problém ani mladší děti, které zvládají ovládání počítače myší nebo touchpadem. Hra funguje i na mobilních přístrojích s dotykovým displayem. Testována byla na sedmipalcovém tabletu s operačním systémem Android 4.2.2 v prohlížeči Mozilla Firefox pro Android.

Pravidla jsou stejná, jako u mladšího karetního předchůdce. Tedy nalezení stejné dvojice obrázků. Mohla by tedy zabavit i děti kolem tří let, které poznají stejné obrázky. Samozřejmě s někým, kdo jim pomůže aplikaci na zařízení ovládat.

### 4.3.1 *Přístupnost a možnosti rozšíření*

Průběh hry je doprovázen zvukem, tudíž by mohla být i vhodná pro některé osoby se zrakovým postižením, které mají alespoň zbytky zraku a rozliší barvy na monitoru.

Aby byla hra ještě přístupnější dala by se rozšířit o popis obrázků (tzv. alt), který by mohl pomoci čtečkám obrazovek. Nebyla však možnost, otestovat tento předpoklad.

Dále by mohl být přidán i zvukový signál, při přejetí myši přes jednotlivé karty (např. čísla dle umístění karet), aby osoba, která hru hraje, věděla, že se kurzor nachází v prostoru pro kliknutí a tudíž otočení obrázku.

## 5 ZÁVĚR

Touto bakalářskou prací jsem chtěl hlavně poukázat na novinky a možnosti tvorby v novém standardu HTML5 a s ním souvisejících kaskádových stylech a JS.

Nové elementy společně s novou verzí CSS a použití jazyka JS obohatily tento standard o nové možnosti tvorby a vdechly webu a webovým aplikacím nový život.

V první kapitole této práce jsem objasnil rozdíl mezi webovými, nativními a hybridními aplikacemi. Upřesnil, jak fungují a jaké jsou možnosti jejich tvorby.

V druhé kapitole jsem se zaměřil na hlavní a důležité elementy jazyka HTML5, strukturu samotného dokumentu, vlastnosti elementů a jejich praktické použití.

Ve třetí části této práce jsem představil, co uživatel potřebuje pro tvorbu takové aplikace a jaké nástroje by k tomu mohl využít. Na konci jsem zmínil, jaké omezení provází tento standard, a problémy s ním spjaté.

Poslední část jsem věnoval samotné aplikaci, kterou jsem vytvořil za pomoci elementů značkovacího jazyka HTML5, kaskádových stylů a programovacího jazyka JavaScript. Shrnujím v této kapitole všechny předchozí poznatky a prakticky ukázal, co se například dá vytvořit ve webovém prostředí. Jedná se o jednoduchou hru na motiv klasické karetní hry PEXESO, kde byly využity všechny možnosti tvorby zmíněné v této práci.

Samotné HTML5 vdechuje nový život do webových aplikací. Element canvas v jistých ohledech a možnostech nahrazuje klasické flash animace a aplikace a umožňuje tak uživateli, programátorovi jednoduše vytvořit vlastní webovou aplikaci, pouze za pomoci webového prohlížeče a libovolného textového editoru.

Doufám, že mnou vytvořená hra PEXESO a popis jejího kódu, někomu pomůže, nebo ho inspiruje k vlastní tvorbě hry, nebo aplikace. Rád bych, aby mé poznatky a informace umístěné v této práci, pomohly čtenáři v řešení jeho problému, nebo k nalezení potřebné informace.

## 6 SEZNAM ZDROJŮ

1. PETR, J. Multikanálová budoucnost: jaké mobilní aplikace budete vytvářet vy? *ITBIZ Vaše jednička mezi nulami* [online]. 25. 11. 2013 [cit. 2014-03-18]. Dostupné z: <http://www.itbiz.cz/clanky/multikanalova-budoucnost-jake-mobilni-aplikace-budete-vytvaret-vy>
2. MICHL, P. Je lepší nativní aplikace nebo mobilní web? *Marketing Journal - marketing, public relations, reklama, internet* [online]. 25. 6. 2012 [cit. 2014-03-18]. Dostupné z: [http://www.m-journal.cz/cs/internet/je-lepsi-nativni-aplikace-nebo-mobilni-web-\\_\\_s281x9241.html](http://www.m-journal.cz/cs/internet/je-lepsi-nativni-aplikace-nebo-mobilni-web-__s281x9241.html)
3. ČÁPKA, D. 1. díl - Úvod do PHP a webových aplikací. *Devbook.cz* [online]. [cit. 2014-03-22]. Dostupné z: <http://www.devbook.cz/php-tutorial-uvod-do-webovych-aplikaci>
4. CASTELDINE, E. M. EFTOS a M. WHEELER. *Vytváříme mobilní web a aplikace pro chytré telefony a tablety..* Překlad Jakub MUŽÍK. Brno: Computer Press, 2013, 288 s.. ISBN 978-80-251-3763-5.
5. KORF, M. a E. OKSMAN. Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options. *Developer Force: Salesforce.com* [online]. [cit. 2014-03-18]. Dostupné z: [http://wiki.developerforce.com/page/Native,\\_HTML5,\\_or\\_Hybrid:\\_Understanding\\_Your\\_Mobile\\_Application\\_Development\\_Options](http://wiki.developerforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options)
6. Adobe Dreamweaver \* Webové aplikace. *Adobe* [online]. [cit. 2014-03-22]. Dostupné z: [http://help.adobe.com/cs\\_CZ/dreamweaver/cs/using/WS51D271ABC466-4e03-B8B7-8ADE9D3B67DEa.html](http://help.adobe.com/cs_CZ/dreamweaver/cs/using/WS51D271ABC466-4e03-B8B7-8ADE9D3B67DEa.html)
7. *Mozilla Developer Network* [online]. [cit. 2014-03-24]. Dostupné z: <https://developer.mozilla.org/>
8. ŠKULTÉTY, R. *JavaScript: Programujeme internetové aplikace*. 2. vyd. Brno: Computer Press, 2004, 224 s.. ISBN 80-251-0144-4.
9. BAŠE, O. *jQuery pro neprogramátory: Průvodce využitím knihovny jQuery UI*. Brno: Computer Press, 2012. ISBN 978-80-251-3750-5.
10. SUEHRING, S. a J. VALADE. *PHP, MySQL, JavaScript & HTML5 All-in-One For Dummies*. New Jersey: John Wiley & Sons, Inc. 2013. ISBN 978-1-118-21370-4.
11. FREEMAN, E. a ROBSON. *Head First HTML5 Programming*. Sebastopol: O'Reilly Media, 2011, 573 s.. ISBN 978-1-449-39054-9.
12. *W3Schools Online Web Tutorials* [online]. [cit. 2014-04-26]. Dostupné z: <http://www.w3schools.com>

13. HTML5 - World Wide Web Consortium [online]. [cit. 2015-03-24]. Dostupné z: <http://www.w3.org/TR/html5/>
14. On SGML and HTML. *The World Wide Web Consortium (W3C)* [online]. [cit. 2014-04-15]. Dostupné z: <http://www.w3.org/TR/REC-html40/intro/sgmltut.html>
15. LUBBERS, P. B. ALBERS a F. SALIM. *HTML5: Programujeme moderní webové aplikace*. Computer Press, a.s. 2011. ISBN 978-80-251-3539-6.
16. WAI-ARIA 1.0 User Agent Implementation Guide. *The World Wide Web Consortium (W3C)* [online]. [cit. 2014-04-26]. Dostupné z: <http://www.w3.org/TR/wai-aria-implementation/>
17. *GIMP* [online]. [cit. 2015-03-18]. Dostupné z: <http://www.gimp.org/about/introduction.html>
18. Audacity: O Audacity [online]. [cit. 2015-04-07]. Dostupné z: <http://audacity.sourceforge.net/about/>
19. LAIRD, C. Unleash the Power of Hardware-Accelerated HTML5 Canvas [online]. [cit. 2015-04-20]. Dostupné z: <http://www.htmlgoodies.com/html5/client/unleash-the-power-of-hardware-accelerated-html5-canvas.html#fbid=HRFMQRprVg1>
20. HALLEM, S. Overcoming HTML5's Limitations [online]. 13. 8. 2013 [cit. 2015-04-05]. Dostupné z: <http://www.drdobbs.com/web-development/overcoming-html5s-limitations/240159696>
21. KOVAŘÍKOVÁ, B. Kdo vynalezl PEXESO? [online]. [cit. 2015-04-07]. Dostupné z: <http://www.bkovarikova.cz/texty/zajimavosti/kdo-vynalezl-pexeso.htm>
22. Hybridní aplikace. *Vývoj mobilních aplikací - Pixelfield s.r.o.* [online]. [cit. 2014-03-22]. Dostupné z: <http://www.vyvojmobilniaplikace.cz/hybridni-aplikace/>
23. BACON, J. *Practical PHP and MySQL : building eight dynamic web applications*. Boston (USA): Pearson Education, 2007, 512 s.. ISBN 0-13-223997-3.

## 7 SEZNAM OBRÁZKŮ A TABULEK

|                                                                                                                                                     |    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|----|
| OBRÁZEK 1 – ROZDĚLENÍ JEDNOTLIVÝCH ARCHITEKTUR DLE JEJICH SCHOPNOSTÍ PRACOVAT OFFLINE<br>A ZÁVISLOSTI NA PLATFORMĚ (PŘEVZATO A UPRAVENO Z [5])..... | 8  |
| OBRÁZEK 2 – ODKAZY A ZNAČKY VŠECH TŘÍ DOSTUPNÝCH TRŽIŠŤ.....                                                                                        | 9  |
| OBRÁZEK 3 – SCHÉMA KOMUNIKACE MEZI WEBOVÝM PROHLÍŽEČEM A WEBOVÝM SERVEREM.....                                                                      | 13 |
| OBRÁZEK 4 – SCHÉMA KOMUNIKACE MEZI WEBOVÝM PROHLÍŽEČEM A WEBOVÝM SERVEREM<br>ZPRACOVÁVAJÍCÍM DYNAMICKOU STRÁNKU (PŘEVZATO Z [6]) .....              | 14 |
| OBRÁZEK 5 – PŘÍKLAD KOMUNIKACE MEZI INTERNETOVÝM PROHLÍŽEČEM, WEBOVÝM A DATABÁZOVÝM<br>SERVEREM (PŘEVZATO A UPRAVENO Z [21]) .....                  | 15 |
| OBRÁZEK 6 – JEDNODUCHÝ FORMULÁŘ V INTERNETOVÉM PROHLÍŽEČI (ZDROJ: AUTOR) .....                                                                      | 16 |
| OBRÁZEK 7 – VÝVOJOVÉ PROSTŘEDÍ MACROMEDIA FLASH 7 (ZDROJ: AUTOR).....                                                                               | 22 |
| OBRÁZEK 8 – ROZLOŽENÍ STRÁNKY POMOCÍ NOVÝCH ELEMENTŮ .....                                                                                          | 24 |
| OBRÁZEK 9 – SOUŘADNICE PLÁTNÁ .....                                                                                                                 | 28 |
| OBRÁZEK 10 – PŘÍKLAD PODPORY ELEMENTU AUDIO (10/2014).....                                                                                          | 32 |
| OBRÁZEK 11 - PROSTŘEDÍ PROGRAMU ADOBE PHOTOSHOP CC (ZDROJ: SNÍMEK OBRAZOVKY).....                                                                   | 35 |
| OBRÁZEK 12 - PRACOVNÍ PROSTŘEDÍ PROGRAMU GIMP 2.8 (ZDROJ: SNÍMEK OBRAZOVKY) .....                                                                   | 36 |
| OBRÁZEK 13 – PRACOVNÍ PROSTŘEDÍ PROGRAMU AUDACITY (ZDROJ: SNÍMEK OBRAZOVKY) .....                                                                   | 39 |
| OBRÁZEK 14 – ČASOVÁ OSA PRVNÍCH WEBOVÝCH PROHLÍŽEČŮ .....                                                                                           | 40 |
| OBRÁZEK 15 – VÝVOJÁŘSKÉ NÁSTROJE PROHLÍŽEČŮ MOZILLA FIREFOX (VLEVO) .....                                                                           | 41 |
| OBRÁZEK 16 – HRA BEJEWELED (ZDROJ: SNÍMEK OBRAZOVKY).....                                                                                           | 42 |
| OBRÁZEK 17 – QUAKE V ELEMENTU CANVAS HTML5 (ZDROJ: SNÍMEK OBRAZOVKY) .....                                                                          | 43 |
| OBRÁZEK 18 – ROBOTS ARE PEOPLE TOO (ZDROJ: SNÍMEK OBRAZOVKY) .....                                                                                  | 43 |
| OBRÁZEK 19 – OBRÁZKY POUŽITÉ V PEXESU VČETNĚ RUBOVÉ STRANY (ZDROJ: SNÍMEK OBRAZOVKY) ....                                                           | 47 |
| OBRÁZEK 20 – PROSTŘEDÍ HRY (ZDROJ: SNÍMEK OBRAZOVKY) .....                                                                                          | 48 |
| OBRÁZEK 21 – POSTUPNÉ VYKRESLENÍ VŠECH KARET NA PLÁTNĚ (ZDROJ: AUTOR) .....                                                                         | 53 |
| <br>TABULKA 1 – PODPORA FORMÁTŮ SOUBORŮ ELEMENTU <AUDIO> V JEDNOTLIVÝCH PROHLÍŽEČÍCH [12]                                                           | 30 |
| TABULKA 2 – TŘI POUŽÍVANÉ FORMÁTY A JEJICH KÓDOVÁNÍ [11].....                                                                                       | 30 |

## **8 SEZNAM PŘÍLOH**

**Příloha 1** 1x CD s praktickou částí práce – hrou PEXESO

## **Příloha 1**

---

## ANOTACE

|                          |                                        |
|--------------------------|----------------------------------------|
| <b>Jméno a příjmení:</b> | Roman Krymlák                          |
| <b>Katedra:</b>          | Katedra technické a informační výchovy |
| <b>Vedoucí práce:</b>    | Mgr. Jan Kubrický, Ph.D.               |
| <b>Rok obhajoby:</b>     | 2015                                   |

|                                    |                                                                                                                                                                                                                                                                                                                                     |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Název práce:</b>                | Tvorba interaktivních her pro internetové prohlížeče                                                                                                                                                                                                                                                                                |
| <b>Název v angličtině:</b>         | Creation interactive games for web browsers                                                                                                                                                                                                                                                                                         |
| <b>Anotace práce:</b>              | Bakalářská práce se zaměřuje primárně na webový standard HTML5, JavaScript a využití jejich možností při tvorbě webové aplikace (hry). V práci jsou popsány vlastnosti jednotlivých elementů, které jsou následně prakticky využity v jednoduché webové hře PEXESO.                                                                 |
| <b>Klíčová slova:</b>              | HTML5, JavaScript, webová aplikace, hra                                                                                                                                                                                                                                                                                             |
| <b>Anotace v angličtině:</b>       | This bachelor's thesis is primarily concerned with the HTML5 web standard, JavaScript and the utilization of their facilities during the process of creating a web application (a game). The characteristics of the individual elements are described in the thesis and afterwards practically applied to a simple web game PEXESO. |
| <b>Klíčová slova v angličtině:</b> | HTML5, JavaScript, web application, game                                                                                                                                                                                                                                                                                            |
| <b>Přílohy vázané v práci:</b>     | Příloha – 1x CD s praktickou částí práce – hrou PEXESO                                                                                                                                                                                                                                                                              |
| <b>Rozsah práce:</b>               | Celkový počet stran 61, z toho přiložené přílohy 1 strana                                                                                                                                                                                                                                                                           |
| <b>Jazyk práce:</b>                | Český jazyk                                                                                                                                                                                                                                                                                                                         |