

**Jihočeská univerzita v Českých Budějovicích**  
**Přírodovědecká fakulta**



**Tvorba aplikace Pixwords**  
**v prostředí Firefox OS**

**Bakalářská práce**

**Radek Hiess**

**Vedoucí práce: Martin Stránský, Red Hat**

**Garant: Mgr. Jiří Pech, PhD.**

**České Budějovice 2016**

**Jihočeská univerzita v Českých Budějovicích**  
Přírodovědecká fakulta

**ZADÁVACÍ PROTOKOL BAKALÁŘSKÉ PRÁCE**

**Student: Radek Hiess**  
(jméno, příjmení, tituly)

**Obor – zaměření studia: 1801R001 / Aplikovaná informatika**  
.....

**Školitel: Martin Stránský, Red Hat Brno, stransky@redhat.com**  
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

**Garant z PřF: Mgr. Jiří Pech, PhD., UAI**  
(jméno, příjmení, tituly, katedra – jen v případě externího školitele)

**Školitel – specialista, konzultant: .....**  
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

**Téma bakalářské práce: Tvorba aplikace Pixwords v prostředí Firefox OS**

Cíle práce :

- 1) Popište prostředí operačního systému Firefox OS. Porovnejte s ostatními mobilními platformami jako např. Android OS.
- 2) Vytvořte aplikaci typu Pixwords pro prostředí Firefox OS. Použijte k tomu technologie HTML5 a JavaScript. K vývoji použijte desktopový prohlížeč Firefox s integrovaným simulátorem. Aplikace musí podporovat Firefox OS ve verzi minimálně 1.2. Aplikace musí podporovat různé rozlišení, minimálně však 320x480 pixelů.
- 3) Otestujte vaši aplikaci v integrovaném simulátoru a také přímo na zařízení s operačním systémem Firefox OS.
- 4) Zhodnoťte vytvořenou aplikaci a porovnejte s aplikacemi dostupnými pro jiné mobilní platformy.

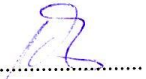
Základní doporučená literatura :

Financování práce : .....

Vedoucí práce: Martin Stránský ..... podpis : 

U externích vedoucích fakultní garant práce Mgr. Jiří Pech, Ph.D. podpis : 

Garant oboru bakalářského studia  
..... podpis : .....

Vedoucí oddělení: PhDr. Milan Novák, Ph.D. ..... podpis : 

Případný souhlas vedoucího ústavu AV ..... podpis : .....

V Českých Budějovicích dne ..... 25.11. 2014 .....

Převzal/a dne ..... podpis : 

## **Bibliografické údaje**

Hiess R., 2016: Tvorba aplikace Pixwords v prostředí Firefox OS.  
[Pixwords application development for Firefox OS. Bc. Thesis, in Czech.] – 63 p.,  
Faculty of Science, The University of South Bohemia, České Budějovice,  
Czech Republic.

## **Anotace**

Bakalářská práce seznamuje čtenáře s operačním systémem Firefox OS, popisuje metodiku vývoje aplikací pro tuto platformu, její specifika a použité technologie. Praktická část práce se zabývá analýzou, návrhem a tvorbou mobilní aplikace. Tato aplikace je inspirována populární logickou hrou PixWords. V závěru práce jsou vyhodnoceny výsledky testování a následně je aplikace porovnána s originální hrou PixWords.

## **Annotation**

This bachelor thesis acquaints readers with the operating system Firefox OS. The thesis describes a methodology of development of the application for this platform, its characteristics and the technology used. The practical part deals with an analysis, design and creation of the mobile application. This application is inspired by the popular logic game PixWords. In the conclusion, the results of testing are evaluated and the application is compared to the original game PixWords.

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 22. 4. 2016

Podpis: .....

## **Poděkování**

Děkuji Martinu Stránskému za cenné rady a připomínky a děkuji Mgr. Jiřímu Pechovi, PhD. za odborný dohled. Velký dík patří také mé rodině, která mě po celou dobu studia podporovala.

# Obsah

Úvod.....	1
Cíle .....	2
<b>1. Popis platformy Firefox OS .....</b>	<b>3</b>
1.1. Popis systému .....	3
1.2. Architektura systému .....	4
1.3. Grafické uživatelské rozhraní systému .....	5
1.4. Vývoj aplikací.....	8
1.5. Distribuce aplikací .....	11
1.6. Porovnání s jinými mobilními platformami.....	12
<b>2. Výběr aplikace.....</b>	<b>14</b>
2.1. Situace na trhu .....	14
2.2. Popis aplikace PixWords.....	15
<b>3. Metodika vývoje.....</b>	<b>17</b>
3.1. Zvolená metodika .....	17
3.2. Použité technologie a nástroje .....	18
3.3. Použité obrázky.....	23
<b>4. Analýza řešení .....</b>	<b>24</b>
4.1. Funkční požadavky .....	24
4.2. Nefunkční požadavky .....	25
4.3. Diagram případů užití .....	26
4.4. Struktura obrazovek .....	27
4.5. Návrh grafického uživatelského rozhraní .....	27
<b>5. Implementace .....</b>	<b>29</b>
5.1. Grafické uživatelské rozhraní aplikace.....	29
5.2. Aplikační logika.....	30
<b>6. Testování aplikace.....</b>	<b>37</b>
6.1. Testování během vývoje .....	37
6.2. Uživatelské testování .....	39
<b>7. Zhodnocení aplikace .....</b>	<b>42</b>
7.1. Srovnání aplikací .....	42
7.2. Zhodnocení aplikace .....	42
7.3. Budoucí vývoj.....	43
<b>Závěr.....</b>	<b>44</b>
<b>Seznam použité literatury.....</b>	<b>45</b>
<b>Seznam obrázků .....</b>	<b>49</b>
<b>Seznam příkladů.....</b>	<b>49</b>
<b>Přílohy .....</b>	<b>49</b>

# Úvod

Chytrá zařízení zažívají v současné době čím dál větší rozmach. Se vzrůstajícím zájmem uživatelů o moderní technologie se výrobci snaží vymýšlet nový hardware i software a vydobýt si tak své místo na trhu. Právě o toto se v roce 2013 pokusila také společnost Mozilla, kdy zveřejnila Firefox OS. Firefox OS je operační systém s otevřeným zdrojovým kódem určený pro chytrá zařízení. Aplikace pro Firefox OS se vyvíjí jako HTML5 aplikace s využitím JavaScriptu a tzv. Web API. Díky tomuto Web API mohou aplikace komunikovat přímo s hardwarem zařízení.

Inspirace pro dané téma plyne ze zájmu autora seznámit se s touto novou platformou, vyzkoušet si její možnosti a tvorbu kompatibilní aplikace. Cílem bakalářské práce je vytvořit a otestovat logickou hru se slovy a obrázky podobnou populární hře PixWords. Práce se snaží srozumitelným způsobem předat budoucím vývojářům zajímavým se o Firefox OS informace o systému, jakož i další poznatky potřebné k vývoji Firefox OS aplikací. Dokončená práce tak kompenzuje absenci česky psané literatury o této platformě a zaplňuje díru na trhu s aplikacemi.

První část práce pojednává o samotném systému, jeho prostředí, vysvětluje základní pojmy a porovnává systém s dalšími platformami. Dále je zdůvodněn výběr aplikace a vysvětlena metodika a technologie použité při vývoji. Stěžejní pasáž práce se zabývá analýzou, návrhem řešení a následnou implementací uživatelského prostředí a klíčových funkcí aplikace. Aplikace je ve finále testována z hlediska funkčního i z hlediska uživatelského a na základě porovnání s aplikací PixWords zhodnocena.



*Obrázek 1: Ukázka hotové aplikace*



# Cíle práce

- Představit prostředí operačního systému Firefox OS a porovnat jej s platformami Android a iOS. Popsat specifika vývoje aplikací pro tuto platformu a použité technologie.
- Vytvořit za použití technologií HTML5 a JavaScript aplikaci typu Pixwords pro prostředí Firefox OS. Aplikace musí podporovat Firefox OS ve verzi minimálně 1.2 a různá rozlišení displeje. Minimální podporované rozlišení bude 320×480 pixelů.
- Otestovat aplikaci ve webovém prohlížeči Firefox s integrovaným simulátorem a také přímo na zařízení s operačním systémem Firefox OS. Provést uživatelské testy použitelnosti a testování vyhodnotit.
- Zhodnotit vytvořenou aplikaci a porovnat ji s podobnými aplikacemi dostupnými pro jiné mobilní platformy.

# 1. Popis platformy Firefox OS

První kapitola se zabývá teoretickými poznatky o platformě. Je zmíněno, co Firefox OS je, jaká je jeho architektura a jaké má grafické uživatelské rozhraní. Popisuje problematiku a specifika vývoje pro tento systém, druhy aplikací, možnosti distribuce a srovnává prostředí Firefox OS s jinými mobilními systémy.

## 1.1. Popis systému

Firefox OS je open-source operační systém vyvíjený společností Mozilla a několika dalšími přispěvateli. Také se lze setkat s názvem Boot to Gecko (B2G), což je označení projektu mezi vývojáři. Základem systému je Linux kernel a vykreslovací jádro Gecko, které je použito také ve stejnojmenném webovém prohlížeči Mozilla Firefox. [1]

Počátky systému Firefox OS sahají do roku 2011, kdy byl vedením Mozilly oznámen vznik projektu Boot to Gecko. Po více než roce a půl od první zmínky o projektu, v únoru 2013, Mozilla veřejně představila první verzi svého nového operačního systému. První zařízení s tímto systémem bylo uvedeno do prodeje v červenci 2013. Jednalo se o chytrý telefon ZTE Open. V následujících měsících se na trhu objevila další zařízení, jako např. Alcatel One Touch Fire nebo LG Fireweb. Rozšíření systému mezi uživatele měla zajistit spolupráce Mozilly se třemi výrobci mobilních zařízení a sedmnácti mobilními operátory po celém světě. [2] Z prvotních devíti států systém do dnešního dne expandoval do více než třiceti zemí světa, včetně České republiky. [3].

Díky linuxovému jádru je Firefox OS kompatibilní s velkým množstvím chytrých zařízení. Pod pojmem chytrá zařízení si lze představit elektronická zařízení, často multifunkční s operačním systémem, která komunikují s uživatelem, příp. i mezi sebou a která mohou do jisté míry fungovat autonomně. Systém byl původně určen pro chytré telefony s nízkým výkonem, avšak je dodáván také v tabletech, chytrých televizích či jednodeskových počítačích. Má nízké nároky na výkon a mobilní zařízení s Firefox OS disponují relativně malou hardwarovou konfigurací, což se odráží na jejich ceně. Právě nízkou cenou těchto mobilních telefonů Mozilla cílí na trh v Asii, Latinské Americe a Africe. [2]

## 1.2. Architektura systému

Architektura operačních systémů se logicky dělí do několika vrstev. Nejvyšší vrstva je ta, se kterou pracuje uživatel, naopak nejnižší vrstva je postavena nejbližší k hardwaru zařízení. Architektura Firefox OS se skládá ze tří vrstev: [4]

1. Gaia – uživatelské prostředí platformy,
2. Gecko – běhové prostředí aplikací,
3. Gonk – jádro systému.

### 1.2.1. Gaia

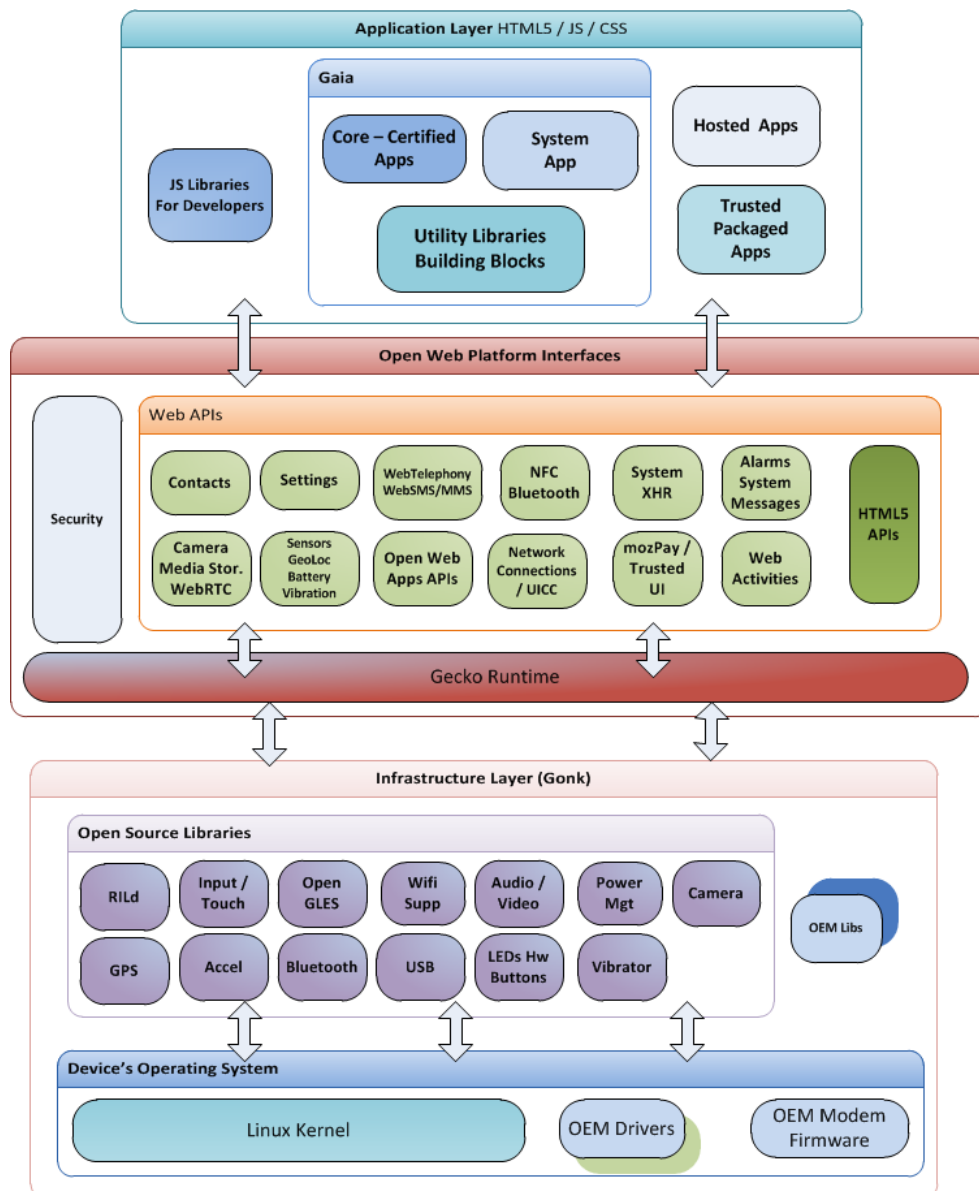
Gaia je uživatelské rozhraní platformy. Jsou to základní systémové aplikace, které zajišťují použitelnost zařízení, např. domovská obrazovka, správce oken, seznam kontaktů či přehrávač videa. Tyto tzv. certifikované aplikace mají plná práva přistupovat ke všem funkcím zařízení. Vedle systémových aplikací běží ostatní nainstalované aplikace.

### 1.2.2. Gecko

Aplikace pro Firefox OS se vyvíjí jako webové aplikace za použití standardů HTML, CSS a JavaScript. Kód aplikace se nekompiluje do spustitelných souborů, ale soubory s kódem jsou zpracovávány v Gecku, které kód interpretuje a vykreslí aplikaci na displej. Součástí Gecka je také tzv. Web API, které zajišťuje komunikaci mezi aplikací a nižší vrstvou Gonk.

### 1.2.3. Gonk

Nejnižší postavenou vrstvou je Gonk. Tato vrstva se skládá ze samotného linuxového jádra, které je založeno na Android Open Source Project, a hardwarové abstrakční vrstvy. Android Open Source Project je otevřený zdrojový kód systému Android. Tento kód může kdokoli převzít a vytvořit z něj vlastní operační systém. Hardwarová abstrakční vrstva je soubor knihoven, které přijímají požadavky od Web API a prostřednictvím ovladačů výrobce komunikují přímo s hardwarem zařízení. [4]



Obrázek 2: Architektura systému [4]

### 1.3. Grafické uživatelské rozhraní systému

Každý systém se od svého počátku vyvíjí a zdokonaluje. S každou novou verzí přichází nové funkce a možnosti. Často se stává, že nová verze operačního systému přinese upravené, nebo dokonce úplně nové uživatelské rozhraní. Výjimkou není ani Firefox OS, jehož grafické uživatelské rozhraní (GUI) se liší nejen podle toho, zda jde o verzi pro telefony, nebo pro televizory, ale více či méně se liší i jednotlivé verze pro daná zařízení. Protože se práce zabývá vývojem mobilní aplikace, bude rozebráno mobilní prostředí.

Grafické uživatelské rozhraní v chytrých telefonech se drží zavedených standardů. Díky tomu je práce se systémem intuitivní a lze tak předpokládat, že i běžný uživatel nebude mít s ovládáním větší problémy.

Mobilní zařízení s Firefox OS, ani systém samotný, nemají navigační tlačítko Zpět. Zařízení disponují pouze tlačítkem pro návrat na domovskou obrazovku. Pokud je potřeba v rámci aplikace přecházet mezi dílčími obrazovkami, musí být tlačítko Zpět řešeno softwarově. Stěžejním prvkem aplikací je proto horní lišta, ve které je vlevo umístěno tlačítko Zpět nebo Zrušit, uprostřed název aplikace nebo dílčí obrazovky aplikace a vpravo jedno či dvě tlačítka pro další možnosti.

Napříč celým systémem je dostupná stahovací lišta s notifikacemi. Notifikační lišta je řešena podobně, jako v systému Android od Googlu. Kromě upozornění v ní najdeme také tlačítka pro zapnutí/vypnutí některých funkcí a další ovládací prvky.

V systémových aplikacích je použité písmo Fira Sans, které bylo navrženo pro Firefox OS tak, aby zapadlo do koncepce systému a současně bylo čitelné na displejích různé velikosti. [5] Fira Sans má otevřenou licenci a může jej využívat kdokoli. Vývojáři mají také k dispozici doporučení, jak písmo používat v aplikacích.

### **1.3.1. První verze Firefox OS**

Domovská obrazovka Firefox OS pro chytré telefony se ve verzích 1.0-1.4 podobá domovské obrazovce Androidu. V horní části obrazovky je pole pro vyhledávání, které vyhledává aplikace v zařízení i obsah na webu. Ve spodní části je statická lišta s ikonami systémových aplikací, např. Kontakty, Zprávy. Ikony systémových aplikace mají kulatý tvar. Všechny nainstalované aplikace jsou vidět přímo na domovské obrazovce, není zde žádná nabídka aplikací ve druhé úrovni. Seznam aplikací je rozdělen po několika ikonách a listuje se v něm horizontálně posunutím vlevo, resp. vpravo.

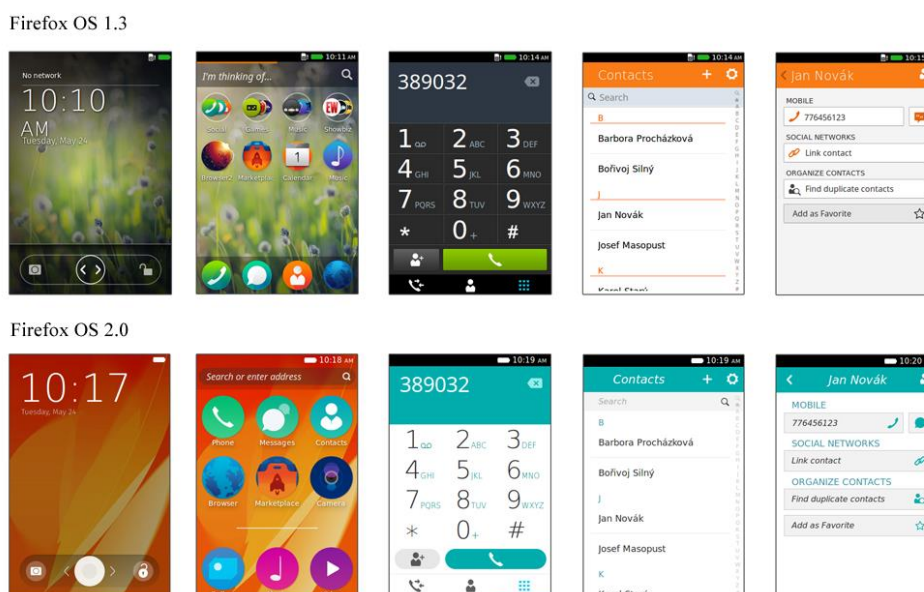
Na domovskou obrazovku je možné vkládat i tzv. Chytré sbírky. Firefox OS automaticky rozřazuje nainstalované aplikace do tematicky podobných kategorií, to jsou Chytré sbírky. Po otevření Chytré sbírky se uživateli zobrazí nainstalované aplikace a relevantní aplikace, které by se mu mohly líbit.

Zamykací obrazovka má stejné pozadí, jako domovská. V horní části zamykací obrazovky je datum a čas. Dole je prvek pro odemčení systému. Posunutím tlačítka doleva se systém odemyká, posunutím doprava se otevře fotoaparát.

### 1.3.2. Druhá verze Firefox OS

Firefox OS prošel velkou úpravou a ve verzi 2.0 přichází s novým grafickým uživatelským rozhraním. To více využívá možnosti plochého designu a je přehlednější. Rozložení prvků ve většině systémových aplikací zůstalo zachováno, některé aplikace však byly přepracovány do světlejších barev. Např. číselník pro vytočení telefonního čísla, který byl původně laděný do tmavě šedé barvy s bílým písmem, byl přebarvený na bílou s černým písmem. Prvky, které byly v první verzi oranžové, dostaly ve druhé verzi příjemnou tyrkysově modrou barvu. To se týká navigační lišty v hlavičce systémových aplikací, podnadpisů či některých tlačítek.

Od verze 2.0 je předělané rozložení ikon na domovské obrazovce. Seznamem aplikací se roluje vertikálně. Ikony v seznamu aplikací jsou větší, umístěné pod sebou ve skupinách. Skupiny aplikací jsou od sebe odděleny bílou čarou a v této verzi nejdou shlukovat. Možnost shluknout a zmenšit skupiny se objevila až ve verzi 2.2. Tuto funkci uživatel ocení v momentě, kdy má na domovské obrazovce větší množství aplikací. Další novinkou je přidání vyhledávacího pole do notificační lišty, což mimo jiné usnadňuje přístup k Internetu.



Obrázek 3: Porovnání GUI Firefox OS 1.3 vs. 2.0

## 1.4. Vývoj aplikací

Pro vývoj aplikací pro Firefox OS je potřeba znát standardní webové technologie, které se používají při vyvážení běžných webových stránek. Těmito technologiemi jsou HTML, CSS a JavaScript. Zevrubný popis technologií je v kapitole 3.2. Jednoduchost a otevřenost těchto technologií otevírá dveře webovým vývojářům, kteří nemají tolik zkušeností s psaním nativních aplikací v programovacích jazycích Java či C#. Právě od rozšíření těchto technologií si Mozilla slibovala rychlý nárůst počtu kompatibilních aplikací.

### 1.4.1. Minimální požadavky

Každá aplikace pro Firefox OS se musí skládat alespoň z:

- jednoho souboru ve formátu HTML,
- manifestu aplikace,
- bitmapové ikony o velikosti 128×128 pixelů.

Chceme-li aplikaci nabízet v oficiálním obchodě Firefox Marketplace, musíme připojit také snímek obrazovky za běhu aplikace. Typická aplikace je však složitější a sestává tak z manifestu, několika souborů se zdrojovými kódy, různých velikostí ikony, obrázků, zvuků apod. [6]

### 1.4.2. Manifest aplikace

Manifest aplikace je jedna z klíčových věcí, které odlišují aplikaci pro Firefox OS od běžné webové stránky. Je to souhrn základních informací o aplikaci, které jsou potřeba k instalaci a spuštění aplikace. Je v něm definován např. název, popis, ikony, verze aj. Obsahuje také seznam Web API, která aplikace vyžaduje, či typ aplikace (webová/privilegovaná/certifikovaná). Pro manifest platí tyto tři konvence: název souboru je *manifest.webapp*, soubor je umístěn v kořenovém adresáři aplikace, zápis manifestu je ve formátu JSON (JavaScript Object Notation). [7]

### 1.4.3. Web API

Samotné aplikace nemají přístup k hardwaru zařízení. Proto existuje Web API (Application Programming Interface). To je zabezpečené rozhraní, které webovým aplikacím zajišťuje přístup k funkcím hardwaru zařízení a k uloženým datům. Zjednodušeně se dá říct, že když chce vývojář např. zjistit polohu pomocí GPS, vystačí si s jednoduchou funkcí v JavaScriptu. O všechno ostatní se postará Web API a hardwarová abstrakční vrstva. [4]

Ve skutečnosti se jedná o sadu několika rozhraní, která je možné dle funkce kategorizovat do čtyř skupin: [8]

- komunikační API (WiFi Information API, WebSMS aj.),
- API pro přístup k hardwaru (Geolocation API, Camera API aj.),
- API pro správu dat (IndexedDB, Device Storage API aj.),
- ostatní API (Alarm API, Web Activities aj.).

Z bezpečnostních důvodů nemají všechny aplikace právo přistupovat ke všem rozhraním. Web API jsou tedy rozdělena z hlediska bezpečnosti na: [9]

- obecná – dostupná pro všechny aplikace,
- privilegovaná – dostupná pouze pro privilegované aplikace,
- certifikovaná – dostupná pouze pro interní aplikace.

### 1.4.4. Typy aplikací

V předchozím odstavci bylo řečeno, že většina aplikací nemůže používat všechna Web API. Je nežádoucí, aby aplikace, která je nainstalovaná z neznámého zdroje, mohla pracovat např. s kontakty v telefonu. Firefox OS proto rozlišuje, na základě způsobu instalace a přidělených práv, dva základní typy aplikací: hostované aplikace a balené aplikace.

#### **Hostované aplikace**

Hostovaná aplikace je uložena na webovém serveru, stejně jako webové stránky. Při prvním připojení k serveru a nainstalování aplikace se manifest aplikace uloží do zařízení. Ostatní data se do zařízení neukládají a při dalším spuštění aplikace se načítají znovu ze vzdáleného serveru.



V bezpečnostním modelu se hostované aplikace dají přirovnat k běžným webovým aplikacím. To znamená, že nemohou používat privilegovaná ani certifikovaná Web API. [10]

### **Balené aplikace**

Balená aplikace je soubor ve formátu ZIP, který obsahuje manifest aplikace a veškerá data potřebná pro běh. Nainstalovaná aplikace (včetně dat) je uložena v zařízení a spustí se i bez připojení k internetu.

Existují tři typy balených aplikací: webové aplikace, privilegované aplikace, interní aplikace. Každý z těchto typů odpovídá jedné úrovni v bezpečnostním modelu. [11]

### **Webové aplikace**

Toto je výchozí úroveň všech aplikací. Jde o obyčejné webové aplikace, které nevyužívají privilegovaná nebo certifikovaná Web API.

### **Privilegované aplikace**

Privilegované aplikace jsou takové, které pracují s některým z privilegovaných Web API. Lze je přirovnat k nativním aplikacím, které známe např. ze systému Android. Musí projít speciálním schvalovacím procesem na oficiálním obchodě Firefox Marketplace. Jakmile je aplikace schválena, Firefox Marketplace opatří manifest aplikace digitálním podpisem. [9]

### **Interní aplikace**

Interní (certifikované) aplikace jsou ty, které využívají certifikovaná Web API. Certifikovaná (interní) Web API poskytují přístup ke kritickým funkcím systému. Interní aplikace nejsou dostupné přes Firefox Marketplace, ale musí být ověřené výrobcem či distributorem zařízení.

## 1.5. Distribuce aplikací

Když je aplikace hotová, pravděpodobně ji budeme chtít ukázat světu. Také k možnostem distribuce má Mozilla vcelku liberální postoj. Už podle dělení aplikací na hostované a balené se dá usoudit, že aplikace můžeme do zařízení instalovat z několika různých zdrojů.

### 1.5.1. Distribuce svépomocí

Aplikace můžeme distribuovat svépomocí z vlastní domény. K tomu je zapotřebí, aby aplikace měla implementovanou funkci, která zavolá manifest a aplikaci nainstaluje. Pokud chceme z vlastní domény instalovat balenou aplikaci, musíme vedle balíku s aplikací přidat tzv. mini-manifest. Mini-manifest slouží k instalaci a následné aktualizaci balené aplikace.

Kromě distribuce na vlastní doméně se nabízí využít některý z neoficiálních marketů. Ty fungují na podobném principu jako oficiální Firefox Marketplace, avšak stále neřeší hlavní nevýhodu. Nevýhodou aplikací distribuovaných svépomocí je to, že nemohou využívat privilegovaná Web API. [12]

### 1.5.2. Distribuce prostřednictvím Firefox Marketplace

Firefox Marketplace je oficiální internetový obchod Mozilly, kde vývojáři nabízejí svoje aplikace pro Firefox OS, Firefox a Firefox pro Android. Jedná se o hlavní distribuční kanál pro Firefox OS aplikace.

Pro publikování aplikací na Firefox Marketplace je nutné být přihlášený pod účtem vývojáře. Registrovaní vývojáři pak mají možnost přidávat a spravovat aplikace či zobrazovat statistiky stahování, hodnocení aj. Přidání aplikace uživatelem je rychlé a intuitivní. Každá aplikace odeslaná do marketu musí projít schvalovacím procesem, ten může trvat až několik dní. Na hostované aplikace jsou kladeny mírnější schvalovací nároky, než na privilegované, protože nevyužívají citlivá Web API. [13]

## 1.6. Porovnání s jinými mobilními platformami

### 1.6.1. Firefox OS vs. Android

Android je operační systém vyvíjený společností Google. Jak již bylo zmíněno, Android má otevřený kód a jeho základem je jádro Linuxu. Přestože Firefox OS vychází ze stejného jádra, architektura těchto systémů se liší.

Zatímco Firefox OS má třívrstvou, systémová architektura Androidu má pět vrstev: aplikace, aplikační rámec (application framework), systémové služby, hardwarová abstrakční vrstva, jádro Linuxu. [14] V závislosti na architektuře se liší také typy aplikací, vývojářské nástroje apod. Aplikace pro Android se programují v jazyce Java. Z tohoto důvodu je součástí systému kompilátor, který aplikace překládá do strojového kódu.

Aby vývojář mohl vytvářet aplikace pro Android, potřebuje k tomu vývojové prostředí a Android Software Development Kit. Android SDK je sada softwarových nástrojů obsahující např. debugger, knihovny, emulátor systému aj. [15] Při vytváření aplikací pro Firefox OS si naopak tvůrce vystačí s vývojovým prostředím a Firefox OS Simulátorem, který se instaluje jako doplněk do webového prohlížeče Mozilla Firefox.

Rozdíly jsou také v distribuci aplikací. Chceme-li publikovat Android aplikace v oficiálním obchodě Google Play, musíme zaplatit jednorázový registrační poplatek 25 USD. [16] Zveřejnění aplikace v Firefox Marketplace je zdarma. Určitou výhodou pro koncové uživatele může být to, že pro instalaci aplikace z Firefox Marketplace se nemusí přihlašovat do žádného účtu.

### 1.6.2. Firefox OS vs. iOS

iOS je operační systém unixového typu od společnosti Apple. Výjma několika open-source komponent je iOS proprietární software dodávaný výhradně v zařízeních Apple. Architektura systému má čtyři vrstvy: Cocoa Touch, Media, systémové služby, jádro systému. [17] Vývoj a distribuce aplikací jsou oproti Firefox OS podstatně odlišné. Aplikace pro iOS jsou nativní, píšou se v jazycích Swift nebo Objective-C (případně s využitím C, C++). Pro tvorbu je potřeba iOS Software

Development Kit, který poskytuje potřebné knihovny. Apple také nabízí vlastní vývojové prostředí XCode, které zahrnuje iPhone SDK či iOS Simulator a je zaměřené na vytváření aplikací pro tuto platformu. [18]

Možnosti distribuce jsou v porovnání s Firefox OS omezené. Aplikace je možné šířit pouze přes oficiální obchod App Store, eventuálně v rámci speciálních programů pro podniky a testery. Pokud chce vývojář zveřejňovat aplikace v App Store, musí být zaregistrován v programu pro vývojáře. Registrovaní vývojáři platí poplatek 99 USD ročně. [19]

## 2. Výběr aplikace

Na následujících řádcích je shrnuto, jak probíhal výběr vyvíjené aplikace, jaké typy aplikací připadaly v úvahu a proč byla vybrána hra PixWords. V kapitole je také uvedena charakteristika a princip této hry.

### 2.1. Situace na trhu

Na úplném začátku byla provedena analýza, jaké Firefox OS aplikace již existují. Záměrem analýzy bylo zvolit specifické zaměření aplikace. Přehled dosavadních aplikací byl sestaven s využitím oficiálních obchodů: Firefox Marketplace, Google Play, App Store. Porovnávána byla především nabídka těchto obchodů.

Situace je taková, že drtivá většina vývojářů cílí na masově užívané platformy. Např. hra PixWords je dostupná pro Android, iOS, Windows Phone i BlackBerry OS, ale takováto rozšířenost je spíše vzácností. Převážná část mobilních aplikací je dostupná jen pro Android a iOS.

Vybírány byly především takové aplikace, u kterých se dalo s jistotou očekávat, že je uživatelé ocení a budou je používat. Nakonec se do úzkého výběru dostaly dva naprosto odlišné druhy softwaru: FTP klient a hra PixWords. FTP klient je program, který umí manipulovat se soubory na vzdáleném zařízení přes File Transfer Protocol.

PixWords není jediná hra, jejíž podstatou je rozpoznat slovo podle obrázku. Další oblíbený koncept je *4 obrázky 1 slovo*. V tomto typu her má hráč k dispozici čtyři obrázky a z nabízených písmen musí složit slovo, které vystihuje, co mají tyto obrázky společné. Jiný podobný koncept spočívá v hledání více slov v jednom společném obrázku.

Z analýzy vyplynulo, že pro Firefox OS je v marketu pouze jedna hra takového typu. Jmenuje se *4 Images 1 Word*. Tato hra však není příliš propracovaná. Má jen základní uživatelské rozhraní bez jakýchkoli grafických úprav a ovládání také není příliš pohodlné. FTP klient, jako samostatná aplikace pro Firefox OS, existuje. Jeho grafické uživatelské rozhraní je prosté, ale podporuje všechny funkce potřebné pro práci s FTP. Oblíbenost hry PixWords na jiných platformách je opravdu značná, takže je pravděpodobné, že si najde dostatek fanoušků i mezi uživateli Firefox OS.

Z těchto důvodů převážilo rozhodnutí vytvořit mobilní aplikaci podobnou hře PixWords. Aplikace dostala název **Foxword**.

## 2.2. Popis aplikace PixWords

PixWords je logická hra vydaná společností Dekovir Entertainment, ve které má hráč za úkol doplňovat slova do křížovky, přičemž hledané slovo si musí odvodit z přiloženého obrázku. Některá slova jsou jednoznačná. Naopak u jiných je opravdu náročné vydedukovat, co obrázek vyjadřuje. Na obrázku bývají znázorněna rozmanitá témata – osoby, zvířata, věci, činnosti, vlastnosti apod. Hledaná slova nemusí být jen podstatná jména, ale i slovesa či jiné slovní druhy.

O velké popularitě hry svědčí nejen vysoký počet uživatelů, kterých jsou po celém světě miliony. Rozluštit některá slova je náročné a to bylo impulsem k tomu, že si hráči začali vyměňovat správné odpovědi na sociálních sítích. Existuje také hodně mobilních aplikací a webových stránek, na kterých lze neznámá slova rychle a jednoduše vyhledat.

Přidanou hodnotou této hry je, že si hráč zdokonalí svoji slovní zásobu a to nejen v rodném jazyce. Dle široké nabídky jazyků a obrovského množství slov, se kterými se uživatel seznámí, se na PixWords můžeme dívat jako na didaktickou hru vhodnou pro procvičování cizích jazyků.

### 2.2.1. Princip hry

PixWords má více než 700 úrovní, v každé úrovni je křížovka s prázdnými políčky pro čtyři slova. Při vybrání požadované oblasti slova se zobrazí nabídka písmen, která lze použít k vytvoření tohoto slova. Když hráč správně doplní některé ze čtyř slov, odhalí písmeno jiného slova a na virtuální konto se mu přičte jedna mince. Doplní-li správně všechna slova v dané úrovni, přejde na další. Pro případ, že by si hráč nevěděl rady s některým slovem, má k dispozici tři nápovědy: odkrytí prvního písmene slova; odkrytí náhodného písmene; odstranění písmen, která jsou v nabídce navíc. Za využití nápovědy se však z virtuálního konta strhávají poměrně velké částky mincí. Má-li hráč na virtuálním kontě málo mincí, a přesto by chtěl použít nápovědu, nabízí se několik možností, jak mince nabýt. Buď si může mince

dokoupit za reálné peníze, nebo je může získat jinou formou, např. za zhlédnutí videa, stažení aplikace apod.

Při zakládání nové hry je nutné zvolit, v jakém jazyce budou hledaná slova. Na výběr je přes 40 jazyků. Není problém mít rozehráno více her současně, každou v jiném jazyce. Hráč jistě ocení, že virtuální konto s mincemi je mezi rozehranými hrami sdílené. Díky tomu může mince nasbírané např. ve hře s českými slovy zúročit ve hře anglické.

## 3. Metodika vývoje

Tato kapitola rozebírá, jaký postup byl při tvorbě zvolen. Vysvětluje metodiku, nástroje a zdroje využití při vývoji. Seznamuje čtenáře s použitými technologiemi a knihovnou třetí strany.

### 3.1. Zvolená metodika

Tvorba programu nebo informačního systému je komplexní proces, který se dále dělí a řeší specifické podproblémy. Z tohoto důvodu vznikly různé metodiky vývoje, tj. doporučený souhrn etap, postupů a nástrojů, který pokrývá celý životní cyklus projektu. [20]

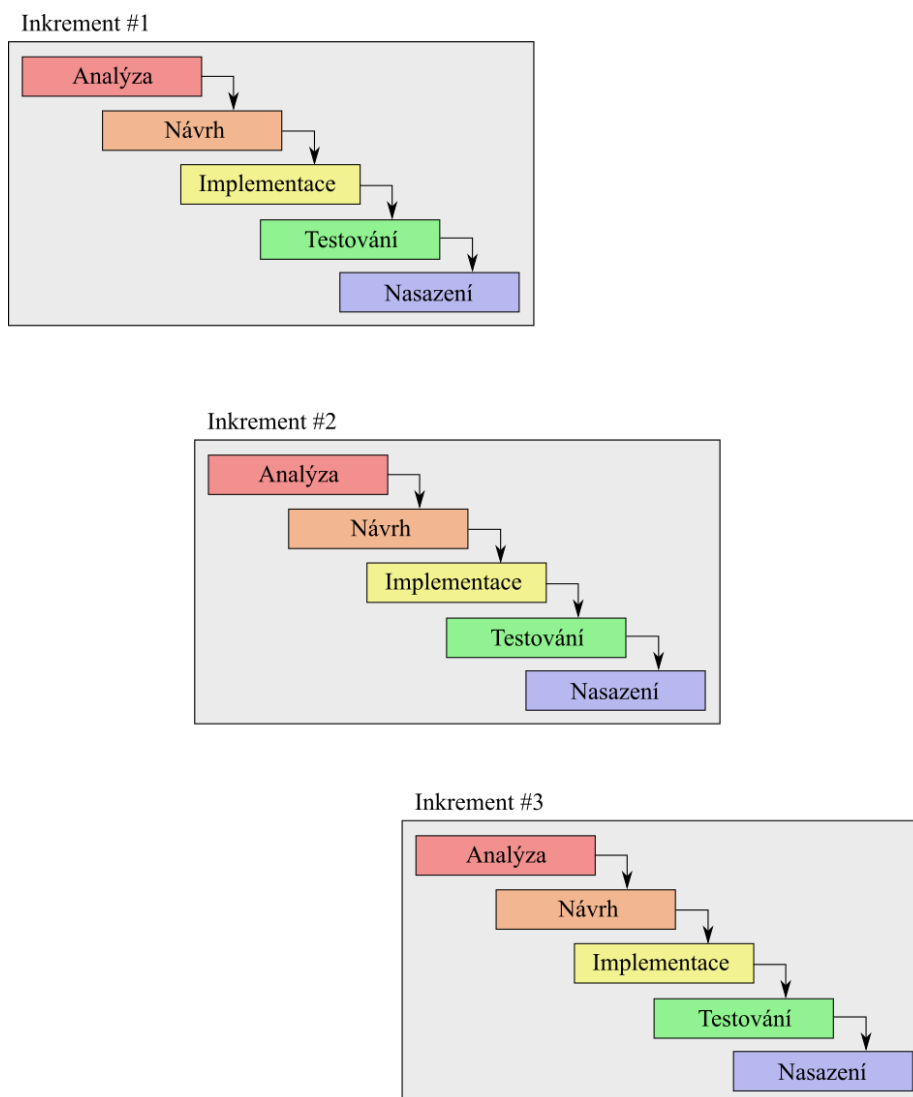
Za klasický sekvenční vývojový model je považován tzv. Vodopád. Vodopádový model zpracovává projekt, jako lineární posloupnost fází (specifikace požadavků, návrh, implementace, testování, údržba), z nichž každá musí být před přechodem do další fáze formálně dokončena a ověřena. [21]

Po zvážení požadavků, rizik a možností byl vzhledem k povaze aplikace pro vývoj zvolen inkrementální model. V inkrementálním modelu vývoje se využívá iterativní přístup, avšak projekt je rozdělen na několik menších etap – inkrementů, které se řeší na principu již zmíněného vodopádového modelu. Každý inkrement se pak skládá z několika fází: analýza, návrh, implementace, testování, nasazení. V prvních fázích inkrementu se vybere a navrhne funkce s nejvyšší prioritou. Poté se vybraná funkce implementuje a otestuje. Na konci inkrementu se hotová funkce integruje do celku a přechází se na další inkrement. Tento proces se opakuje, dokud nejsou implementovány všechny požadované funkce. Na rozdíl od klasického sekvenčního přístupu má inkrementální přístup tyto přínosy:

- můžeme reagovat na změnu požadavků,
- máme možnost kontrolovat stav projektu mezi jednotlivými inkrementy,
- na konci každého inkrementu máme funkční prototyp.

Na druhou stranu, postupně nabalované funkce mohou způsobit nečekané problémy v architektuře a nestabilitu celého projektu.





Obrázek 4: Inkrementální model vývoje

## 3.2. Použité technologie a nástroje

### 3.2.1. HTML – HyperText Markup Language

Pro kódování uživatelského prostředí aplikace se používá značkový jazyk HTML, resp. jeho modernější verze HTML5. HTML elementy tvoří základní stavební prvky webových stránek/aplikací a vymezují strukturu dokumentu. HTML5 je pro tvorbu webových aplikací vhodnější, než předchozí verze HTML 4.01, protože zavádí velké množství nových funkcí. Vedle nových značek je to lepší práce s multimediálním obsahem, vektorovou grafikou a také podpora práce s lokálním úložištěm zařízení.

### 3.2.2. CSS – Cascading Style Sheets

Souběžně s HTML se používá CSS – kaskádové styly. Kaskádové styly popisují vzhled elementů napsaných v HTML, jako třeba jejich barvu, velikost, pozici a další vlastnosti. Ve výsledku pak je oddělen popis vzhledu uživatelského prostředí od jeho struktury. V současné době se používá CSS ve verzi 3, které mimo jiné umí nastavit vlastní typy písma či definovat různé styly v závislosti na velikosti displeje zařízení.

### 3.2.3. JavaScript

Funkční část aplikace se kóduje ve skriptovacím jazyce JavaScript. Na webu se JavaScript užívá pro tvorbu interaktivních prvků. Umí přidávat, měnit či odebírat HTML elementy a jejich obsah, upravovat kaskádové styly nebo vytvářet vlastní události a reagovat na události vyvolané uživatelem (např. kliknutí myši, otočení displeje). Kód psaný v JavaScriptu se většinou interpretuje ve webovém prohlížeči na straně uživatele, ale v některých případech je možné ho spouštět i na straně serveru. Lze v něm psát jak jednoduché skripty, tak složitější aplikace, přičemž tyto aplikace jsou multiplatformní a fungují v každém zařízení s internetovým prohlížečem.

### 3.2.4. Výběr frameworku

Využití frameworku při vývoji má své výhody. Framework vyřeší typické problémy v dané oblasti a ulehčí tím práci programátora. Vzhledem k použité platformě přicházely v úvahu tyto tři nejznámější frameworky pro tvorbu mobilních webových aplikací:

- jQT (jQTouch) – funguje na knihovnách Zepto.js i jQuery, původně je určený pro prohlížeče s jádrem WebKit, má otevřený kód;
- Sencha Touch – částečně je postaven na vlastní knihovně Ext JS, podporuje návrhový vzor Model-View-Controller, aplikace se píšou v JSON zápisu;
- jQuery Mobile.

Při analýze popularity těchto frameworků byly použity dva zdroje. Tím prvním byl vyhledávač Google. S použitím nástroje Google Trends bylo provedeno porovnání, jaký podíl mají jednotlivé frameworky v celkovém objemu vyhledávání na Googlu. Druhým zdrojem byla odborná diskuzní poradna Stack Overflow. [22] Na Stack Overflow jsou pokládány dotazy kategorizovány pomocí štítků. V rámci analýzy byly porovnávány počty založených diskuzních vláken, vztahujících se k zmíněným frameworkům. Z analýzy vyplynulo, že největší popularitě se těší jQuery Mobile. Na druhém místě je Sencha Touch, na třetím jQT.

Práce se Sencha Touch je složitější a vyžaduje to poměrně dost času, než se s ním vývojář sžije. Způsob, jakým se kódují aplikace v jQT se podobá jQuery Mobile. S ohledem na tyto aspekty, silnou uživatelskou základnu a na základě předchozích zkušeností autora byl jako podpůrný nástroj pro vývoj vybrán open-source framework jQuery Mobile.

### 3.2.5. jQuery Mobile

V jQuery Mobile je vytvořena základní kostra uživatelského rozhraní, zajišťuje provázání HTML s JavaScriptem a pomáhá také v aplikační logice. Vychází z populární javascriptové knihovny jQuery, avšak oproti ní je lépe optimalizovaný pro dotyková zařízení. Jeho součástí jsou také předpřipravená témata s hotovými prvky grafického uživatelského rozhraní anebo AJAX (Asynchronous JavaScript and XML) navigace s animovanými přechody mezi stránkami. Využívá nové možnosti HTML5, kdy se HTML5 elementům přidělují speciální atributy (např. *data-role*, *data-theme*). Podle těchto přidělených atributů framework vytvoří výsledné uživatelské prostředí a zobrazí jej uživateli. Tento způsob je velmi elegantní a šetří práci vývojáři. Navíc zajistí, že aplikace vypadá téměř stejně napříč několika mobilními platformami. V logice aplikace se prostřednictvím jQuery Mobile přistupuje k HTML elementům a dynamicky se mění jejich obsah podle chování uživatele.

### 3.2.6. Vývojové prostředí

Výběr vývojového prostředí je čistě na programátorovi, protože aplikace pro Firefox OS lze kódovat v jakémkoli textovém editoru. Není potřeba žádný speciální kompilátor, aplikace jsou interpretovány v druhé vrstvě architektury Gecko.

Pro vývoj aplikace Foxword byl využit textový editor Atom. Atom je svobodný, multiplatformní, open-source software. Nabízí funkce jako zvýraznění syntaxe, automatické dokončování kódu, změnu vzhledu uživatelského prostředí apod. Nesmírnou výhodou je ohromné množství zásuvných modulů, které jsou pro něj dostupné. Základní editor si díky těmto funkcím a modulům můžeme upravit dle vlastní libosti. Právě otevřenost, nabídka rozšíření a možnost personalizace jsou hlavní důvody, proč byl Atom vybrán.

Pro testování a ladění aplikace byl použit webový prohlížeč Mozilla Firefox. Ten má v sobě integrované nástroje pro ladění kódu, náhled v responzivním režimu a další pomůcky. Na testování aplikace se zaměřuje kapitola 6.

### 3.2.7. Responzivní design

Při návrhu grafického uživatelského prostředí aplikace byl jako nejlepší řešení shledán responzivní web design. Responzivní návrh zaručí, že se rozvržení a velikost prvků přizpůsobí velikosti zařízení bez ohledu na to, zda se jedná o malý displej mobilu nebo velkou televizi. Podle Marcotta stojí responzivní design na třech základních pilířích: [23]

1. flexibilní rozvržení,
2. flexibilní obrázky a média,
3. Media Queries.

#### **Flexibilní rozvržení**

Důležité je definovat velikost prvků v relativních jednotkách – procenta, *em*, *rem*. Tím, že blokovým elementům nastavíme šířku v procentech, se nám vytvoří flexibilní rozložení založené na pomyslné mřížce. Velikost této mřížky se procentuálně vypočítá z velikosti zobrazovací plochy (viewportu), v našem případě je to displej. Velikost písma může být v jednotkách *em* či *rem*. Velikost jednotky *em*

se odvíjí od velikosti písma nadřazeného HTML elementu. Oproti tomu, velikost jednotky *rem* vychází z velikosti písma v kořenovém elementu `<html>`, což je obvykle 16 pixelů.

### **Flexibilní obrázky a média**

Aby se obrázky chovaly podobně flexibilně, jako celá mřížka, nesmí být v elementu `<img>` zadány atributy *width* a *height*. Současně by měly mít v CSS nastavenou vlastnost *max-width: 100%*.

### **Media Queries**

Media Queries umožňují používat různé kaskádové styly podle vlastností zařízení. Nejčastěji je potřeba vytvořit styly šité na míru velikosti zobrazovací plochy, avšak sledovaných vlastností je více. Dotazovat se lze na následující: [23]

- šířka a výška zobrazovací plochy,
- šířka a výška displeje zařízení,
- orientace zařízení (na výšku/na šířku),
- poměr šířky a výšky,
- rozlišení displeje,
- barvy displeje.

Můžeme tak formulovat logické dotazy, podle kterých chceme uplatňovat zvláštní CSS pravidla.

### **Metoda Mobile first**

Při psaní responzivního designu bylo postupováno metodou *Mobile first*. Tato metoda je založena na myšlence, že se v kaskádových stylech nejprve styluje rozvržení pro mobilní zařízení. Teprve potom se přes Media Queries styluje rozložení pro větší zobrazovací plochy.

### 3.3. Použité obrázky

Smyslem hry Foxword je uhodnout slovo podle obrázku. V každé úrovni je křížovka se třemi slovy, ke každému slovu se vztahuje jeden obrázek. Z toho vyplývá, že bylo potřeba vytvořit seznam slov, která budou použita, a k němu shromáždit vhodné obrázky. Sběr obrázků probíhal souběžně s vývojem aplikace.

Z pohledu autorského práva jsou použity dvě skupiny obrázků. Většina obrázků je volné dílo (public domain), tzn. dílo, jehož autorská práva nejsou chráněna. Na zbylé obrázky se vztahuje některá z veřejných licencí. Veřejná licence dovoluje ostatním při dodržení určitých podmínek toto dílo dále používat, šířit, případně i upravovat. Příkladem použité veřejné licence je *Creative Commons Uved'te původ 3.0 (CC BY 3.0)*. Tato licence dovoluje dílo užívat, sdílet a upravit pro jakýkoliv účel, a to i komerční, za podmínky, že bude uveden autor díla, odkaz na licenci a budou vyznačeny provedené změny. [24] Pro srovnání je vhodné uvést, že pokud chceme používat či sdílet dílo, jehož autor si vyhrazuje všechna práva, musíme mít výslovný souhlas autora. Aby byly dodrženy podmínky, u každého obrázku v aplikaci je uvedena zkratka licence. Je-li to vyžadováno, je uveden i autor obrázku. Obrázky byly čerpány z různých veřejných webových stránek. Nejčastějším zdrojem je databáze fotografií *Public Domain Images*, viz zdroj [25].

## 4. Analýza řešení

Než se přejde k samotné tvorbě jakékoli aplikace, je potřeba provést alespoň základní analýzu požadavků, funkcionalit a navrhnout adekvátní řešení. Tato část práce se věnuje analýze projektu z hlediska funkčních a nefunkčních požadavků. Na základě definovaných požadavků jsou následně vytvořeny diagramy a návrh uživatelského prostředí.

### 4.1. Funkční požadavky

Funkční požadavky specifikují, jak by se měla aplikace chovat v určitých situacích.

- **Založení nové hry** – aplikace musí umět založit novou hru, musí si pamatovat, v jakém jazyce hra je a na které úrovni se uživatel nachází.
- **Výběr jazyka hry** – aplikace bude při zakládání nové hry nabízet volbu jazyka doplňovaných slov.
- **Podpora několika paralelních her** – aplikace bude umožňovat mít rozehráno více her současně, každou v jiném jazyce. Nebude možné mít rozehráno více her ve stejném jazyce.
- **Zobrazení obrázku ve větší velikosti** – aplikace bude umět přiblížit obrázek hledaného slova a zobrazit jej ve větší velikosti.
- **Zobrazení celkového počtu mincí** – aplikace bude uživateli přidělovat za správně doplněná slova určitý počet mincí. Aplikace si bude pamatovat celkový počet mincí, které hráč nasbíral, a bude mu jej zobrazovat. Mince budou mezi hrami sdílené.
- **Využití nápovědy ve hře** – aplikace bude umožňovat využít nápovědy. Aplikace bude nabízet nápovědy: odkrytí náhodného písmene; odstranění písmen, která jsou v nabídce navíc. Za využití nápovědy bude stržen určitý počet mincí.
- **Možnost nastavit jméno hráče** – aplikace bude umožňovat nastavit jméno hráče.

- **Možnost nastavit obrázek hráče** – aplikace bude umožňovat načíst obrázek z telefonu a nastavit jej jako tzv. avatar hráče.
- **Možnost resetovat hru** – aplikace bude umožňovat začít hrát hru znovu od první úrovně.

## 4.2. Nefunkční požadavky

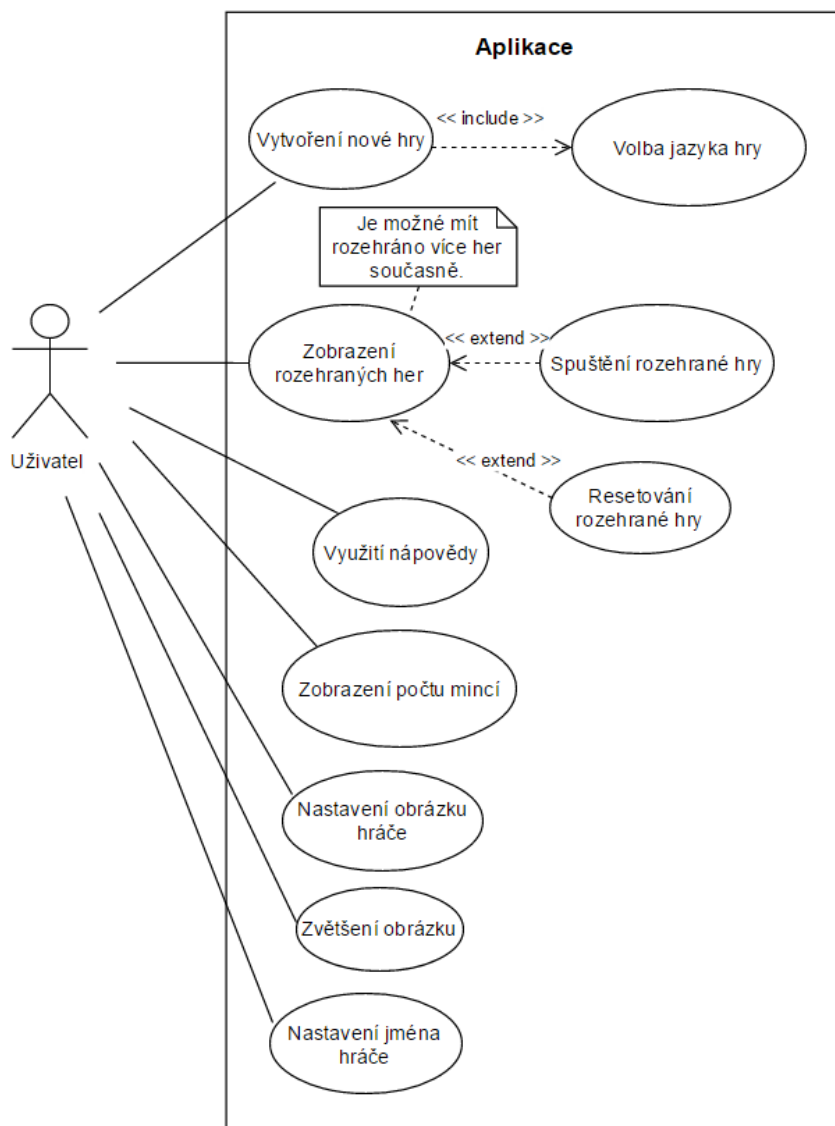
Nefunkční požadavky specifikují vlastnosti a omezení aplikace.

- **Podpora prostředí Firefox OS** – aplikace musí fungovat v systému Firefox OS ve verzi 1.2 nebo novějším.
- **Podpora různých rozlišení** – aplikace musí podporovat různou velikost zobrazovací plochy, minimálně však 320×480 pixelů.
- **Perzistence** – aplikace musí uložit aktuální stav hry a data uživatele do paměti zařízení.
- **Intuitivní ovládání** – aplikace bude navržena tak, aby si uživatel její ovládání rychle osvojil.
- **Rozšiřitelnost o nové jazyky** – aplikace bude umožňovat přidání dalších jazykových verzí.



### 4.3. Diagram případů užití

V závislosti na funkčních požadavcích byl vytvořen diagram případů užití (Use Case Diagram). Ten zobrazuje chování aplikace z pohledu uživatele. Diagram byl vytvořen v on-line nástroji na adrese [www.draw.io](http://www.draw.io).



Obrázek 5: Diagram případů užití

## 4.4. Struktura obrazovek

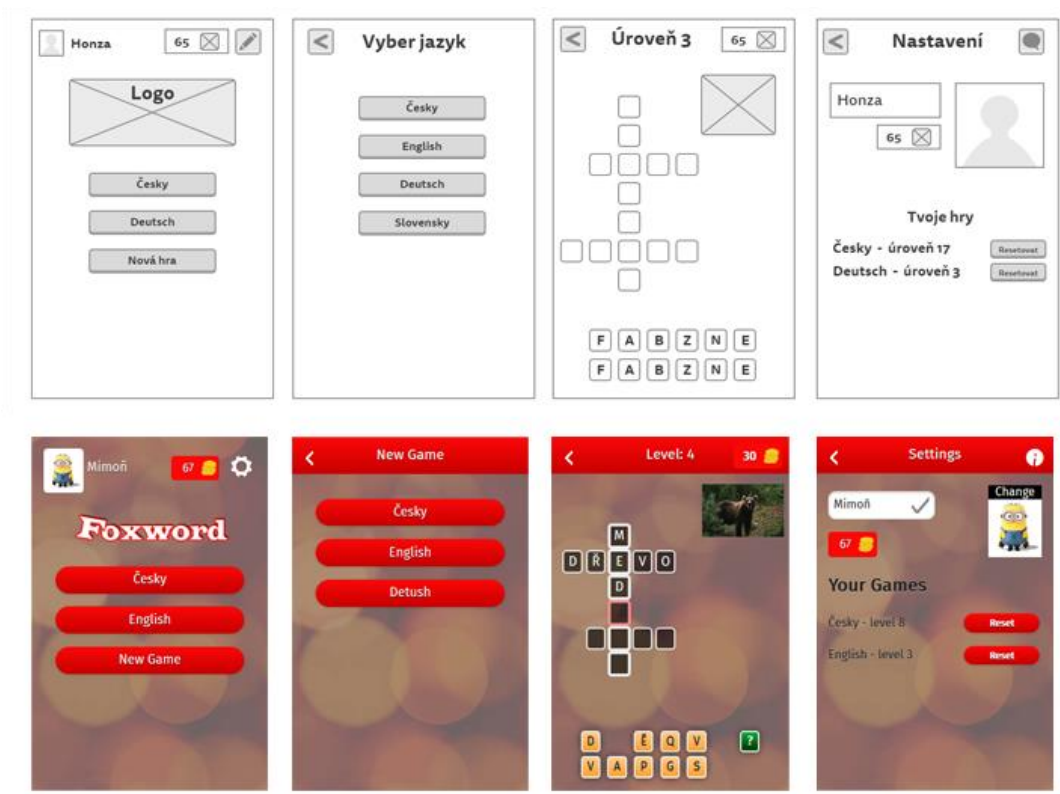
K návrhu hierarchie obrazovek aplikace byla použita jednoduchá myšlenková mapa. Myšlenková mapa byla vytvořena v programu FreeMind, který je volně ke stažení ze zdroje [26].



Obrázek 6: Myšlenková mapa – struktura obrazovek

## 4.5. Návrh grafického uživatelského rozhraní

Když je stanoveno, jaké obrazovky bude aplikace mít, je potřeba navrhnout, jak budou vypadat. Uživatelské rozhraní bylo navrženo pomocí tzv. drátěného modelu (wireframe). Drátěný model definuje rozložení prvků na obrazovce. Zpravidla neřeší barvy, podobu prvků a další detaily, spíše nám pomáhá udělat si lepší představu o struktuře aplikace a její podobě z pohledu uživatele. Na obrázku 6 je drátěný model čtyř hlavních obrazovek vyvíjené aplikace. Na tomtéž obrázku jsou pro porovnání také snímky finální podoby grafického uživatelského rozhraní obrazovek. Zleva je to: úvodní obrazovka, volba jazyka při zakládání nové hry, běžící hra, nastavení. Další snímky obrazovky za běhu aplikace jsou znázorněny v příloze A. Wireframe byl vytvořen ve specializovaném on-line nástroji na adrese [www.gomockingbird.com](http://www.gomockingbird.com).



Obrázek 7: Drátěný model uživatelského rozhraní a snímky obrazovek

## 5. Implementace

V kapitole je vysvětlena implementace důležitých částí aplikace, práce s pamětí zařízení či možnost rozšiřitelnosti o další jazykové verze hry. Některé věci jsou pro lepší pochopení objasněny na příkladech.

### 5.1. Grafické uživatelské rozhraní aplikace

#### 5.1.1. Obrazovky aplikace

Celá struktura grafického uživatelského rozhraní je obsažena v jednom hlavním souboru *index.html*. Obrazovky aplikace a přechod mezi nimi jsou v režii jQuery Mobile. Jednoduchá obrazovka může vypadat třeba takto.

```
<div data-role="page" id="play" data-theme="a">
  <div data-role="header">
    <h1>Level 1</h1>
  </div>
  <div data-role="content">
    <p>Nějaký text</p>
  </div>
</div>
```

*Příklad 1: Definice obrazovky v jQuery Mobile*

Atribut *data-role="page"* zajistí, že se k tomuto prvku bude přistupovat jako k samostatné stránce – obrazovce aplikace. Atribut *data-theme* přiřadí elementu patřičné kaskádové styly, podle nichž bude obarvený.

#### 5.1.2. Mřížka pro vykreslení křížovky

Stěžejním prvkem na herní obrazovce je mřížka o velikosti osm sloupců a devět řádků. V této mřížce jsou připravena políčka křížovky. Každé políčko má atribut *id*, který jej jednoznačně identifikuje. Identifikátor značí pozici políčka v mřížce a je ve formátu *gřs*, kde *ř* je řádek mřížky a *s* je sloupec mřížky. Zde je příklad políčka, které se nachází v mřížce na řádce 4 a ve sloupci 2.

```
<div class="grid-letter" id="g42">&nbsp;</div>
```

*Příklad 2: Políčko křížovky v mřížce*

Políčka jsou ve výchozím stavu skrytá. Pomocí kaskádových stylů a JavaScriptu se vykreslují požadovaná políčka křížovky, do kterých pak hráč doplňuje písmena. Dle velikosti mřížky je maximální délka slov omezena na devět znaků.

## 5.2. Aplikační logika

Aplikační logika je psána v JavaScriptu. JavaScript je sice objektově orientovaný jazyk, ale běžné třídy, které známe z klasických objektově orientovaných jazyků, v podstatě nemá. Existují v něm pouze funkce. Použitím funkce však lze definovat vlastní předpis objektu, přičemž tento předpis je sám sobě konstruktorem. Voláním této funkce s klíčovým slovem *new* se vytváří instance.

### 5.2.1. Objekty

Aplikace Foxword používá dva hlavní objekty: *Word* a *Level*. Klíčovou částí aplikace je objekt *Word*, který definuje slovo a jeho polohu v mřížce. Jeho parametry jsou:

- **name** – slovní identifikátor slova;
- **wordArr** – pole s písmeny slova, názvem příslušného obrázku a zkratkou licence obrázku;
- **row** – číslo řádku v mřížce, od kterého se bude vypisovat slovo;
- **col** – číslo sloupce v mřížce, od kterého se bude vypisovat slovo;
- **orient** – orientace, ve které se bude vypisovat slovo do mřížky (horizontálně, vertikálně);
- **declaredLetters** – nabídka písmen, ze kterých hráč skládá slovo a informace, zda je to písmeno slova, nebo náhodně vygenerované.

Objekt *Word* vypadá následovně:

```
var Word = function () {
  this.name = "";
  this.wordArr = [];
  this.row = 0;
  this.col = 0;
  this.orient = "";
  this.declaredLetters =
    [[], [], [], [], [], [], [], [], [], [], [], [], []];
}
```

*Příklad 3: Objekt Word*

Objekt *Level* definuje úroveň aplikace. V každé úrovni jsou tři slova, takže *Level* obsahuje tři instance objektu *Word*. Vstupním parametrem konstruktoru je aktuální číslo úrovně. Dále obsahuje objekt *doneWordsInLevel*. Tento objekt reprezentuje seznam správně doplněných slov v rozehrané úrovni. Pro každou hru se ukládá vlastní seznam, aby hráč mohl mít rozehráno více her současně.

```
var Level = function (currentLevel) {
  this.currentLevel = currentLevel;
  this.doneWordsInLevel = {};
  this.word0 = new Word();
  this.word1 = new Word();
  this.word2 = new Word();
}
```

*Příklad 4: Objekt Level*

### 5.2.2. Načtení úrovně

Při spuštění každé hry je potřeba vědět, jakou úroveň a jazyk načíst. Každé tlačítko, jež odkazuje na herní obrazovku, má atributy *data-game-lvl* a *data-game-lang*. Hodnotou *data-game-lvl* je číslo úrovně, na které se má hra spustit. Atribut *data-game-lang* v sobě ukrývá kód jazyka hry. Hodnoty obou atributů se načítají z paměti zařízení.

Po přechodu na herní obrazovku se nejprve zjistí hodnoty *data-game-lvl* a *data-game-lang*. Poté se vytvoří instance objektu *Level* a předá se jí číslo úrovně. Při vytváření nové hry je parametr *data-game-lvl* roven nule. Podle kódu jazyka se do pole *levels* načtou z JSON souboru jednotlivé úrovně hry pro daný jazyk. Z paměti zařízení se načtou správně doplněná slova a slova, u nichž byla využita nápověda.

Posléze se zavolá metoda *loadLevel*. Metoda *loadLevel* načte z pole *levels* požadovanou úroveň, postupně naplní objekty *word0*, *word1*, *word2* a zavolá další metody:

- **writeToGrid** – vykreslí do mřížky políčka křížovky s písmeny slova,
- **declareLetters** – vygeneruje sadu písmen, ze kterých hráč skládá slovo, a uloží ji do pole *declaredLetters*.

### 5.2.3. Vložení písmene do křížovky

Ačkoli se to zprvu nezdálo, ukázalo se, že vložení písmene z nabídky do křížovky je nejkompexnější událost v herní logice. Funkce *clickOnBottomOffer* má parametry: *wordOrientation*, *selectedLetter* a *selectedLetterId*. Parametr *wordOrientation* je pole s aktuálně vybraným slovem a jeho orientací v mřížce; proměnná *selectedLetter* vyjadřuje zvolené písmeno, které se má vložit z nabídky do křížovky; proměnná *selectedLetterId* je atribut *id* zvoleného písmene. Průběh funkce po kliknutí na zvolené písmeno ve spodní nabídce je rozebrán v následujícím výčtu.

1. Do proměnné *rightWord* se uloží správná hodnota aktuálně doplňovaného slova. Z písmen, která hráč doplňuje, se postupně vytváří dočasné slovo.
2. Zjistí se atribut *id* políčka v mřížce, do kterého se má zvolené písmeno doplnit.
3. Písmeno zmizí ze spodní nabídky, viditelně se objeví v křížovce.
4. Dočasné slovo se porovná s *rightWord*. Pokud je slovo doplněno chybně, písmena zmizí z mřížky.
5. Pokud je doplněné slovo správné, zobrazí se potvrzovací hláška, zavolá se funkce vibrace, slovo se uloží do paměti zařízení a na virtuální konto se přičtou tři mince.
6. Pokud jsou všechna tři slova doplněna správně, zavolá se vibrace, údaje o hotové úrovni se resetují nebo uloží do paměti zařízení, resetuje se mřížka s křížovkou a načte se další úroveň hry.
7. Pokud dodělaná úroveň byla poslední úrovní hry, resetují se údaje o hře a zobrazí se hláška o úspěšném dokončení.

#### 5.2.4. Náповědy ve hře

Přijít na správné slovo si leckdy vyžádá velké úsilí. V případě, že má hráč s některým slovem potíže, jsou pro něj připraveny dvě nápovědy. První nápověda odhalí v křížovce náhodné písmeno vybraného slova. Druhá skryje z nabídky písmena, která jsou v ní navíc. V nabídce zůstanou jen písmena vybraného slova.

Jakmile hráč zvolí nápovědu, načte se z paměti počet mincí na virtuálním kontě. Jestliže je mincí dostatek, zobrazí se či skryjí písmena, odečtou se mince a do paměti zařízení se uloží informace o použité nápovědě. Využití nápovědy stojí hráče dvacet mincí. Je-li odměna za jedno správně doplněné slovo tři mince, může hráč použít nápovědu průměrně v každé třetí hře.

#### 5.2.5. Ukládání aplikačních dat do paměti zařízení

Hra pracuje s různými daty, která je potřeba ukládat buď dočasně během hraní, nebo perzistentně. Dlouhodobě se ukládají rozehrané hry (jazyk, číslo úrovně, správně doplněná slova), počet získaných mincí, jméno hráče, obrázek hráče. Daty platnými pouze pro rozehranou úroveň jsou uložená slova, u kterých byla využita některá nápověda.

Pro ukládání aplikačních dat se nabízely dvě úložiště: Local Storage a IndexedDB. Zatímco Local Storage umí pracovat pouze s řetězcovými hodnotami, do IndexedDB je možné vkládat také pole či objekty. Úložiště IndexedDB mimo jiné poskytuje asynchronní API, což je hlavní důvod, proč bylo upřednostněno. [27] Díky tomu, že IndexedDB přistupuje k paměti zařízení asynchronně, nečeká se na dokončení operace a neblokuje se vykreslení dokumentu. Jako pomocný nástroj pro přístup k úložišti je využita knihovna *async\_storage.js*. Autorem knihovny je Mozilla, přesněji řečeno tým vývojářů z projektu Boot to Gecko. Tato knihovna používá pro ukládání dat IndexedDB a poskytuje funkce, díky nimž lze s IndexedDB pracovat podobně jednoduše, jako s Local Storage. Příklad 5 ukazuje nejčastěji užívanou konstrukci, která slouží pro práci s uloženými hodnotami. V tomto případě se přičítají mince za správně uhodnuté slovo. Nejdřív se do parametru *coins* načte z paměti počet nasbíraných mincí, provede se požadovaná operace (inkrementace, vypsání nové hodnoty) a poté se nová hodnota *coins* uloží do paměti.



```

asyncStorage.getItem('coins', function(coins) {

    /* za jedno slovo je odměna 5 mincí */
    coins = coins + 5;
    $('.coins').text(coins);

    asyncStorage.setItem('coins', coins, function() {
        console.log('coins stored');
    });
});

```

*Příklad 5: Načtení a uložení hodnoty z/do IndexedDB*

## 5.2.6. Načítání obrázku hráče

Aby bylo dosaženo lepšího uživatelského prožitku, má hráč možnost nastavit si vlastní jméno a obrázek. Výběr obrázku z paměti zařízení je zajištěn prostřednictvím tzv. Web Activities. Web Activities slouží k delegování nějaké aktivity na jinou aplikaci. V praxi to vypadá tak, že si uživatel může zvolit, zda chce vybrat obrázek ze zařízení, nebo se vyfotí fotoaparátem. Tato funkce není standardizovaná pro běžný web, ale je podporována pouze na platformě Firefox OS.

Změna avataru se provádí na obrazovce *Nastavení*. Po kliknutí na obrázek se zavolá funkce ze souboru *pickimg.js*, ta vytvoří objekt aktivity, definuje mu, že se má vykonat aktivita *pick*, a že vybírat lze pouze obrázky.

```

var pick = new MozActivity({
    name: "pick",
    data: {
        type: ["image/png", "image/jpg", "image/jpeg"],
        nocrop: true
    }
});

```

*Příklad 6: Výběr obrázku z paměti zařízení*

Funkce pracuje s BLOB (Binary Large Object) objektem souboru, tj. reprezentace binárních dat zvláštním datovým typem. Adresou obrázku je speciální řetězec, který vrací funkce *URL.createObjectURL()*. Původním záměrem bylo ukládat do IndexedDB jen tuto adresu obrázku. Avšak ta se dynamicky mění, takže není možné opětovně zobrazit požadovaný obrázek na jiném místě. Proto bylo zvoleno řešení ukládat celý BLOB. V případě potřeby se z IndexedDB načte celý BLOB objekt a ten nám pak vrátí novou adresu obrázku.

### 5.2.7. Vibrace

Vzhledem k tomu, že aplikace cílí na mobilní zařízení, je aplikována také funkce vibrace. Vibrace podporuje většina mobilních zařízení a uživatelům poskytuje lepší uživatelský prožitek. Spustí se zavoláním funkce `Navigator.vibrate()`, jejímž vstupním parametrem je interval vibrací v milisekundách.

```
/* jedna vibrace po dobu 600 ms */  
window.navigator.vibrate(600);  
  
/* vzor: 80 ms vibruje, 70 ms je pauza, 80 ms vibruje */  
window.navigator.vibrate([80, 70, 80]);
```

*Příklad 7: Funkce vibrace*

### 5.2.8. Jazykové verze hry

Jedním z požadavků na hru je možnost přidávat nové jazykové verze hry. S ohledem na jednoduchou rozšiřitelnost jsou úrovně hry, slova a jejich poloha v mřížce definovány v externím JSON souboru pro každý jazyk zvlášť. Struktura souboru vychází z objektu *Word*. V příloze B je k nahlédnutí ukázka definice první úrovně hry. V dalším JSON souboru je uložena abeceda, ze které se generují písmena do nabídky pro výběr. Název souborů je kód jazyka. Má-li být hra správně spuštěna, musí pro zvolenou jazykovou verzi existovat oba soubory – *levels/cs.json* i *alphabet/cs.json*.

Také obrázky slov jsou v adresářové struktuře rozděleny podle jazykových verzí. Aby se ovšem zabránilo redundanci dat, není vyžadováno, aby pro každou jazykovou verzi hry existovala vlastní sada obrázků. Např. anglická jazyková verze může do hry načítat obrázky z českého adresáře.

Při vytváření nové jazykové verze hry je třeba splnit tyto kroky.

- Vytvořit v adresáři *./json/levels/* soubor s definicí úrovní a slov.
- Vytvořit v adresáři *./json/ alphabet /* soubor s abecedou jazyka.
- Je-li to potřeba, vytvořit v adresáři *./img/pics/* podadresář s obrázky určenými pro novou verzi.
- V souboru *index.html* přidat na obrazovku *Choose Language* tlačítko pro zahájení nové hry. Tlačítko musí mít atributy *data-game-lvl="0"* a *data-game-lang="kod\_jazyka"*.

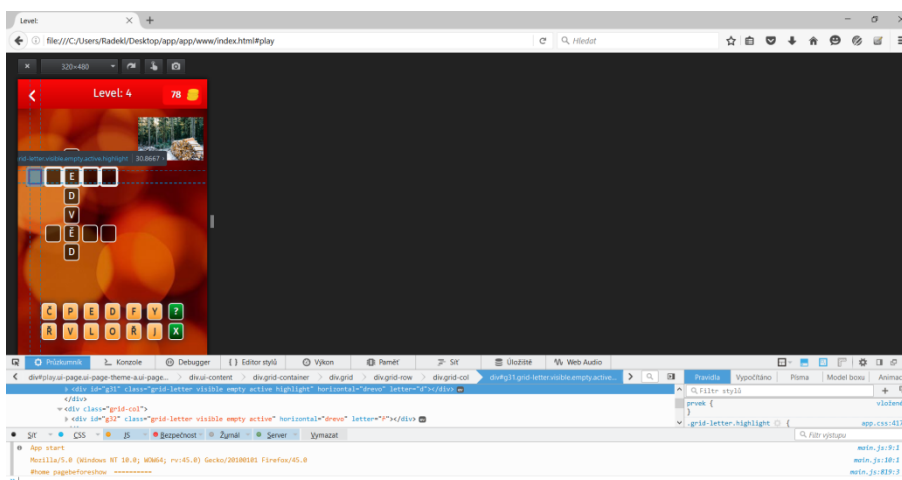
## 6. Testování aplikace

Tato část se zabývá laděním aplikace, testovacími nástroji a stádií testování. Pozornost je věnována jak průběžnému testování funkcí, tak i závěrečnému testování na fyzickém zařízení a testu s reálnými uživateli.

### 6.1. Testování během vývoje

Hlavní fáze testování aplikace spočívala v tzv. testování programátorem. V této fázi testuje aplikaci sám vývojář na úrovni zdrojového kódu. Pro bezprostřední testování a ladění výborně posloužil webový prohlížeč Mozilla Firefox. Prohlížeč k tomuto účelu poskytuje všechny potřebné nástroje. Zde jsou uvedeny ty, které byly využity při vývoji:

- **průzkumník HTML a CSS** – umožňuje sledovat DOM (Document Object Model, tj. stromová, objektově orientovaná reprezentace dokumentu) a kaskádové styly;
- **responzivní režim** – umožňuje měnit velikost zobrazovací plochy;
- **webová konzole** – vypisuje informace o průběhu vykreslení stránky;
- **debugger** – slouží pro ladění chyb v JavaScriptu;
- **průzkumník úložiště** – zobrazí data uložená v lokálním úložišti (Indexed DB, Local Storage).



Obrázek 8: Prohlížeč Mozilla Firefox s ladicími nástroji

### 6.1.1. Firefox OS Simulator

Firefox OS Simulator simuluje běh zařízení se systémem Firefox OS. Je vynikající pomůckou pro testování aplikací, protože umí spustit různé verze Firefox OS s jakýmkoli požadovaným rozlišením zobrazovací plochy. Chování systému i nainstalovaných aplikací v simulátoru je téměř totožné s chováním na fyzickém zařízení. Pouze několik málo Web API nefunguje v simulátoru, protože pro ně v počítači zkrátka chybí hardware (Vibration, WebSMS aj.). Firefox OS Simulator se instaluje jako rozšíření do webového prohlížeče Firefox, takže je multiplatformní.

Testování aplikace v simulátoru bylo provedeno vždy po implementaci nějaké nové funkce, po větší úpravě zdrojového kódu anebo během implementace Web Activities, které neběží ve webovém prohlížeči. Díky simulátoru byla aplikace otestována na těchto verzích systému Firefox OS: 1.1, 1.3, 1.4, 2.0, 2.1, 2.2, 2.6. Aplikace se chová ve všech verzích stejně. Objevil-li se během testování nějaký problém, byl neprodleně vyřešen.

### 6.1.2. Testování na fyzickém zařízení

Hotová aplikace byla na konci vývoje otestována také na fyzickém zařízení s Firefox OS. Pro testování bylo použito mobilní zařízení Sony Xperia Z3 Compact (B2G) s touto konfigurací:

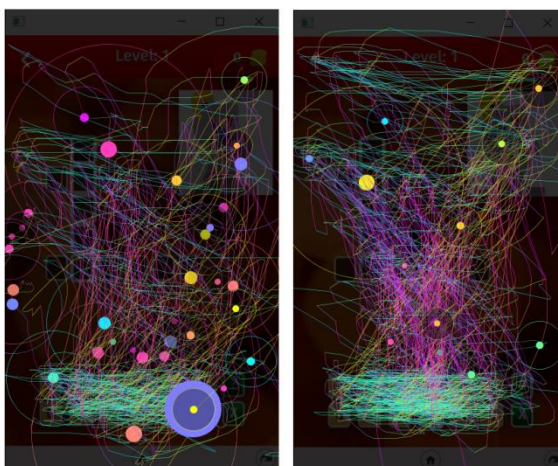
- operační systém: Boot2Gecko 2.6.0.0-prerelease;
- displej: 4,6", 720 × 1 280 pixelů;
- procesor: Qualcomm Snapdragon 801 (4× Krait 400, 2 500 MHz, 28nm);
- paměť RAM: 2 GB.

Aplikace se na mobilním zařízení chovala zcela správně. Taktéž rozložení prvků bylo v pořádku. Přejechy mezi obrazovkami byly plynulé, nepatrnou odezvu vykazovalo načtení a vykreslení obrázku hráče z IndexedDB. Vysoké rozlišení displeje způsobilo, že písmena v křížovce byla malá a zhoršila se tak použitelnost aplikace. Tento nedostatek byl vyřešen zvětšením písma při větší zobrazovací ploše.

## 6.2. Uživatelské testování

Uživatelské testování použitelnosti je důležitou součástí vývoje, protože může odhalit chyby, kterých by si programátor jinak nevšiml. Zejména nám ukáže, jak aplikaci používají běžní uživatelé.

Z důvodu absence mobilního zařízení s Firefox OS bylo testování provedeno na běžném notebooku v Firefox OS Simulatoru. Pro testování bylo simulováno prostředí Firefox OS 2.0 s rozlišením zobrazovací plochy 320×480 pixelů. K ovládání byla použita počítačová myš. Díky tomu, že test probíhal v simulátoru, mohl být na pozadí zapnutý program *IOGraphica* [28], který zaznamenává pohyb myši a vytváří tzv. teplotní mapu. Program sice neumí zachytit klikání tlačítka myši, avšak pro základní ilustraci chování uživatele je dostačující.



Obrázek 9: Ukázka teplotní mapy

### 6.2.1. Průběh uživatelského testování

Vlastního testování se zúčastnilo pět testerů ve věku 22-25 let. Každý samostatně bez přítomnosti ostatních. Tři respondenti uvedli, že již dříve aktivně hráli hru PixWords. Ani jeden z nich neměl předchozí zkušenosti s Firefox OS, ale všichni vlastnili mobilní zařízení s dotykovým displejem a systémem jiného výrobce. Testování probíhalo neřízenou formou. S ohledem na fakt, že testovanou aplikací byla hra, nebyly použity konkrétní testovací scénáře či úkoly.

Nejprve bylo uživateli vysvětleno, o jaký typ aplikace se jedná a jak bude test probíhat. Poté byl posazen k počítači s připraveným simulátorem a dostal

jednoduchý pokyn: *Seznam se s aplikací, prozkoumej její možnosti a hraj si.* V průběhu testování byl každý tester bedlivě pozorován, pozorovatel do postupu testera nijak nezasahoval. Na konci testu byla s uživatelem vedena krátká diskuze, při které mu byly položeny tyto otázky:

- Jak se Ti hra líbí?
- Přijde Ti ovládání pohodlné/intuitivní?
- Co Ti dělalo problémy nebo co bys změnil/a?
- Můžeš aplikaci porovnat s hrou PixWords? Je podle Tebe Foxword lepší, stejný nebo horší než PixWords?
- Vadily Ti anglické popisky uživatelského prostředí?

### **6.2.2. Výsledky uživatelského testování**

Pozorování ukázalo, že většina uživatelů při prvním spuštění aplikace rovnou založila novou hru a začala doplňovat písmena do křížovky. Až po průchodu několika úrovněmi se testeři vraceli na úvodní obrazovku a zkoušeli další možnosti aplikace. Všichni testovaní ovládali aplikaci plynule, bez větších potíží. Doplňování slov do křížovky bylo svižné, občas s využitím nápovědy. Uživatelé, kteří už někdy hráli PixWords, používali gesta naučená z této hry. Naopak nezkušení hráči nepoužívali nápovědu označenou křížkem, nýbrž jen tu s otazníkem.

Z diskuze s testery vyplynulo, že se jim hra líbí a že ovládání aplikace je vcelku intuitivní. Zajímavým zjištěním bylo, že hráčům nevadily anglické popisky tlačítek a obrazovek. Převážná část testovaných si anglických výrazů všimla až na konci testování, kdy na to byli dotázáni. Dva uživatelé ocenili funkci nastavení vlastního obrázku, jeden z nich by uvítal možnost vytvoření více profilů pro různé hráče. Jediný významnější problém byl s doplňováním písmen. Uživatelé si stěžovali, že když bylo některé slovo v křížovce správně doplněno, museli do druhého slova, které jej křížilo, doplnit všechna písmena (tedy i to, které už tam bylo doplněno). Kompletní zápis o chování uživatelů a odpovědi na položené otázky obsahuje příloha C.

Oba zmíněné nedostatky byly opraveny. Náповědy byly sloučeny pod jedno tlačítko s otazníkem. Volba nápovědy by tak měla být srozumitelnější a jednoznačná. Problém s doplňováním křižujících se slov byl vyřešen zvýrazněním aktuálně doplňovaného písmene a přeskočením písmen, která jsou již ve slově doplněna. Tyto úpravy se při opakovaném testování osvědčily.



## 7. Zhodnocení aplikace

Poslední kapitola posuzuje aplikaci, srovnává ji s originální hrou PixWords a hodnotí výsledek vývoje. Ke konci jsou nastíněny další kroky, kterými by se vývoj mohl ubírat.

### 7.1. Srovnání aplikací

Zatímco k vývoji nativních aplikací pro konkurenční operační systémy je zpravidla nutné umět pro každou platformu jiný programovací jazyk, Firefox OS aplikaci vytvoří i webový vývojář se znalostí HTML5 a JavaScriptu. Porovnáme-li vytvořenou aplikaci Foxword s původní předlohou PixWords, najdeme tyto čtyři charakteristické rozdíly:

- Foxword má v křížovce tři slova, PixWords čtyři.
- Foxword nenabízí možnost dokoupit si virtuální mince za reálné peníze.
- Ohodnocení správně doplněných slov mincemi je v Foxword vyšší.
- Cena za nápovědu je v Foxword podstatně nižší, než v PixWords.

Poslední dva zmíněné body platí pro všechny podobné hry s obrázky a slovy (PicWords, 4 Pics 1 Word aj.). Záměrem autora bylo, aby měl uživatel především příjemný požitek ze hry a zlepšil si slovní zásobu. Pokud hráč nemůže uhodnout nějaké slovo, je žádoucí, aby měl dostatek mincí na nápovědu. Protože se nepočítá s dokupováním mincí, je částka za využití nápovědy nízká.

Ke zjištění rozdílů mezi aplikacemi z pohledu uživatele posloužila závěrečná diskuze s testery, jež byla součástí uživatelského testování použitelnosti. Sami hráči potvrdili, že hra je stejná jako PixWords.

### 7.2. Zhodnocení aplikace

Při tvorbě hry byl kladen důraz na otevřenost a hratelnost. Veškeré technologie a nástroje použité při vývoji, od editoru přes knihovny až po testovací simulátor, jsou open-source.

Finální testování neodhalilo žádné závažné chyby v chování aplikace, ani v použitelnosti uživatelského rozhraní. Byly detekovány pouze menší nedokonalosti, které však byly následně odstraněny. Z reakcí získaných při uživatelském testování lze usoudit, že ovládání aplikace je pro hráče pohodlné a hra jako celek jim vyhovuje. S ohledem na tyto skutečnosti lze konstatovat, že se aplikace Foxword povedla a že i s použitím webových technologií lze vytvořit aplikaci, která je konkurenceschopná vůči nativním aplikacím.

Aplikace byla vložena do oficiálního obchodu Firefox Marketplace. Z tohoto distribučního kanálu lze získat další zpětnou vazbu od uživatelů, nápady na zlepšení a v budoucnu tak zhodnotit úspěšnost aplikace z dlouhodobého hlediska.

### **7.3. Budoucí vývoj**

Aplikace je připravena na rozšiřování o nové jazykové verze hry. Přirozeně se tak nabízí implementace dalších jazyků. V nejbližší budoucnosti je v plánu vytvoření hry s anglickými slovy. Vytvoření jiných jazykových mutací by se eventuálně mohla chopit komunita uživatelů okolo Firefox OS.

V současné době je ve hře padesát úrovní s českými slovy. Pokud bude ze strany českých uživatelů zájem, určitě by se přidalo několik dalších úrovní navíc.

Hra již nyní umožňuje nastavit si vlastní jméno a obrázek. Vhodným vylepšením by proto mohlo být propojení se sociální sítí Facebook (nebo i jinou), stažení profilové fotografie hráče z jeho účtu a sdílení výsledků s přáteli.

Další tip na zlepšení zazněl během testování. Uživatel Michal navrhl možnost založit si ve hře více profilů. Hra by si ke každému profilu pamatovala potřebná data a uživatel by měl na výběr, za který profil chce hrát.

Vytvořená aplikace je uvolněna pod licencí GNU GPL (GNU General Public License) jako svobodný software. Do příštího vývoje se tak může zapojit prakticky kdokoli.

# Závěr

Předmětem teoretické části bakalářské práce bylo představit prostředí Firefox OS, porovnat jej s operačními systémy jiných výrobců, vysvětlit základní principy vývoje aplikací a následně popsat zvolený typ vyvíjené aplikace a důvod jejího výběru. Dále je uveden popis zvolené metodiky vývoje či použitých technologií a nástrojů.

Hlavním záměrem práce bylo specifikovat požadavky na aplikaci, navrhnout vhodné řešení a implementovat jej do podoby použitelné mobilní aplikace pro Firefox OS. Nakonec byly nad aplikací provedeny funkční i uživatelské testy a výsledná aplikace byla zhodnocena.

Za použití otevřených webových technologií byla vytvořena funkční open-source hra Foxword a zároveň byl vytvořen ucelený přehled poznatků o tvorbě aplikací pro Firefox OS.

V průběhu psaní bakalářské práce Mozilla přestala nabízet chytré telefony se systémem Firefox OS. Vývoj systému ukončen nebyl. Aktuálně se Mozilla orientuje na chytré televizory a dodává Firefox OS do televizorů Panasonic. Jelikož technologie zůstávají stále stejné, způsob vývoje aplikací pro tuto platformu se nemění. Přestože je aplikace Foxword původně vytvořena pro mobilní zařízení, je responzivní a kompatibilní se všemi zařízeními s Firefox OS. Platí tedy, že si ji můžeme zahrát na televizoru s Firefox OS stejně jako na mobilním telefonu.

# Seznam použité literatury

- [1] Firefox OS. *Mozilla Developer Network* [online]. Mozilla Developer Network, 2012, Last updated: Oct 22, 2015 [cit. 2015-11-04]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Mozilla/Firefox\\_OS](https://developer.mozilla.org/en-US/docs/Mozilla/Firefox_OS)
- [2] Mozilla Announces Global Expansion for Firefox OS. In: *Mozilla Press Center: Mozilla News, Announcements, and More* [online]. Barcelona, 2013 [cit. 2015-11-04]. Dostupné z: <https://blog.mozilla.org/press/2013/02/firefox-os-expansion/>
- [3] Firefox OS devices. *Firefox OS - Devices & Availability - Mozilla* [online]. ©1998-2015 [cit. 2015-11-04]. Dostupné z: <https://www.mozilla.org/en-US/firefox/os/devices/>
- [4] MOZILLA CONTRIBUTORS. Firefox OS architecture. In: *Mozilla Developer Network* [online]. Mozilla Developer Network, 2012, Last updated: Sep 30, 2015 [cit. 2015-11-07]. Dostupné z: [https://developer.mozilla.org/en-US/Firefox\\_OS/Platform/Architecture](https://developer.mozilla.org/en-US/Firefox_OS/Platform/Architecture)
- [5] Firefox OS Typeface. *Mozilla Style Guide* [online]. ©1998-2015 [cit. 2015-12-04]. Dostupné z: <https://www.mozilla.org/en-US/styleguide/products/firefox-os/typeface/>
- [6] RAIDA, Marek. Zkušenosti s psaním Firefox OS aplikací. In: *Zdroják* [online]. Praha: Devel.cz Labs, 2013 [cit. 2015-11-09]. Dostupné z: <https://www.zdrojak.cz/clanky/zkusenosti-s-psanim-firefox-os-aplikaci/>
- [7] MOZILLA CONTRIBUTORS. App manifest. In: *Mozilla Developer Network* [online]. Mozilla Developer Network, 2011, Last updated: Oct 16, 2015 [cit. 2015-11-10]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Mozilla/Firefox\\_OS/Firefox\\_OS\\_apps/Building\\_apps\\_for\\_Firefox\\_OS/Manifest](https://developer.mozilla.org/en-US/docs/Mozilla/Firefox_OS/Firefox_OS_apps/Building_apps_for_Firefox_OS/Manifest)

- [8] MOZILLA CONTRIBUTORS. WebAPI. In: *Mozilla Developer Network* [online]. Mozilla Developer Network, 2012, Last updated: Sep 18, 2015 [cit. 2015-11-11]. Dostupné z: <https://developer.mozilla.org/en-US/docs/WebAPI>
- [9] MOZILLA CONTRIBUTORS. App permissions. In: *Mozilla Developer Network* [online]. Mozilla Developer Network, 2012, Last updated: Sep 7, 2015 [cit. 2015-11-11]. Dostupné z: [https://developer.mozilla.org/en-US/Apps/Fundamentals/App\\_permissions](https://developer.mozilla.org/en-US/Apps/Fundamentals/App_permissions)
- [10] MOZILLA CONTRIBUTORS. Hosted apps. *Mozilla Developer Network* [online]. Mozilla Developer Network, 2014, Last updated: Jan 7, 2015 [cit. 2015-11-14]. Dostupné z: [https://developer.mozilla.org/en-US/Marketplace/Options/Hosted\\_apps](https://developer.mozilla.org/en-US/Marketplace/Options/Hosted_apps)
- [11] MOZILLA CONTRIBUTORS. Packaged apps. In: *Mozilla Developer Network* [online]. Mozilla Developer Network, 2012, Last updated: Aug 31, 2015 [cit. 2015-11-14]. Dostupné z: [https://developer.mozilla.org/en-US/Marketplace/Options/Packaged\\_apps](https://developer.mozilla.org/en-US/Marketplace/Options/Packaged_apps)
- [12] MOZILLA CONTRIBUTORS. Publishing apps yourself. In: *Mozilla Developer Network* [online]. Mozilla Developer Network, 2013, Last updated: Aug 18, 2015 [cit. 2015-11-19]. Dostupné z: [https://developer.mozilla.org/en-US/Marketplace/Options/Self\\_publishing](https://developer.mozilla.org/en-US/Marketplace/Options/Self_publishing)
- [13] The Firefox Marketplace. GARZIA, Andre. *Quick Guide For Firefox OS App Development: Creating HTML5 based apps for Firefox OS* [online]. 0.3. Leanpub, 2014, s. 63-64 [cit. 2015-11-19]. Dostupné z: <https://leanpub.com/quickguidefirefoxosdevelopment>
- [14] The Android Source Code. *Android Open Source Project* [online]. Android Open Source Project [cit. 2015-11-24]. Dostupné z: <https://source.android.com/source/index.html>

- [15] Tools Help. *Android Developers* [online]. Android Open Source Project [cit. 2015-11-24]. Dostupné z: <http://developer.android.com/tools/help/index.html>
- [16] GOOGLE. How to use the Google Play Developer Console. In: *Developer Console Help* [online]. Google, 2015 [cit. 2015-11-24]. Dostupné z: <https://support.google.com/googleplay/android-developer/answer/6112435?hl=en&rd=1>
- [17] APPLE. About the iOS Technologies. In: *IOS Developer Library* [online]. Apple, 2014, Updated: 2014-09-17 [cit. 2015-11-28]. Dostupné z: <https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>
- [18] APPLE. Xcode IDE. *Apple Developer* [online]. Apple, 2015 [cit. 2015-11-28]. Dostupné z: <https://developer.apple.com/xcode/features/>
- [19] APPLE. Choosing a Membership. *Apple Developer* [online]. Apple, 2015 [cit. 2015-11-28]. Dostupné z: <https://developer.apple.com/support/compare-memberships/>
- [20] ŘEPA, Václav. *Analýza a návrh informačních systémů*. 1. vyd. Praha: Ekopress, 1999, 403 s. ISBN 80-86119-13-0.
- [21] WANG, John. *Project management techniques and innovations in information technology*. Hershey, PA: Information Science Reference, c2012. ISBN 978-1-4666-0931-0.
- [22] Tags. *Stack Overflow* [online]. Stack Exchange, 2016 [cit. 2016-01-06]. Dostupné z: <http://stackoverflow.com/tags?tab=popular>
- [23] MARCOTTE, Ethan a [FOREWORD BY JEREMY KEITH]. *Responsive web design*. New York: A Book Apart, 2011. ISBN 978-0-9844425-7-7.
- [24] Uved'te původ 3.0 Česká republika. *Creative Commons* [online]. [cit. 2016-01-11]. Dostupné z: <https://creativecommons.org/licenses/by/3.0/cz/>. Toto je zkrácené, uživatelsky přívětivé znění plného textu licenčních podmínek.

- [25] Public Domain Images [online]. 2016 [cit. 2016-01-11]. Dostupné z: <http://www.public-domain-image.com/>
- [26] Download. FreeMind: free mind mapping software [online]. 2014, last modified on 17 August 2014 [cit. 2016-01-20]. Dostupné z: <http://freemind.sourceforge.net/wiki/index.php/Download>
- [27] MEHTA, Nikunj, Jonas SICKING, Eliot GRAFF, Andrei POPESCU, Jeremy ORLOW a Joshua BELL (eds.). Indexed Database API: W3C Recommendation 08 January 2015. In: W3C [online]. W3C, 2015 [cit. 2016-04-03]. Dostupné z: <https://www.w3.org/TR/2015/REC-IndexedDB-20150108/>
- [28] ZENKOV, Anatoly a Andrey SHIPILO. IOGraphica: Turn your routine work into modern art [online]. Bear On Unicycle, 2016 [cit. 2016-04-11]. Dostupné z: <http://iographica.com>

## Seznam obrázků

Obrázek 1: Ukázka hotové aplikace.....	1
Obrázek 2: Architektura systému [4] .....	5
Obrázek 3: Porovnání GUI Firefox OS 1.3 vs. 2.0 .....	7
Obrázek 4: Inkrementální model vývoje.....	18
Obrázek 5: Diagram případů užití.....	26
Obrázek 6: Myšlenková mapa – struktura obrazovek.....	27
Obrázek 7: Drátěný model uživatelského rozhraní a snímky obrazovek.....	28
Obrázek 8: Prohlížeč Mozilla Firefox s ladícími nástroji .....	37
Obrázek 9: Ukázka teplotní mapy.....	39

## Seznam příkladů

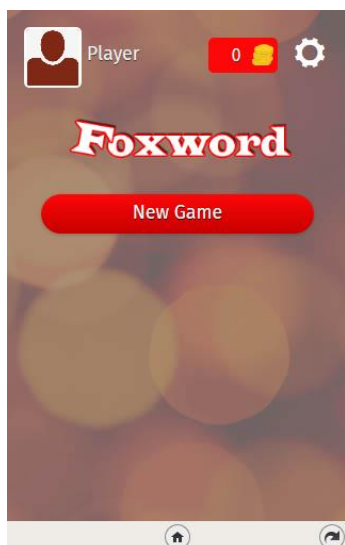
Příklad 1: Definice obrazovky v jQuery Mobile.....	29
Příklad 2: Políčko křížovky v mřížce.....	29
Příklad 3: Objekt Word.....	31
Příklad 4: Objekt Level .....	31
Příklad 5: Načtení a uložení hodnoty z/do IndexedDB.....	34
Příklad 6: Výběr obrázku z paměti zařízení .....	34
Příklad 7: Funkce vibrace.....	35

## Seznam příloh

Příloha A: Snímky obrazovky aplikace.....	50
Příloha B: Definice první úrovně hry .....	52
Příloha C: Zápis z uživatelského testování použitelnosti.....	53
Příloha D: Disk CD .....	63



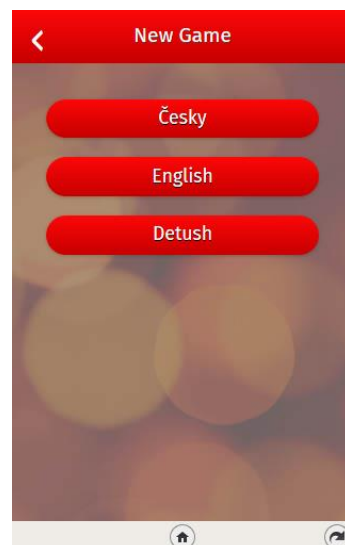
# Příloha A: Snímky obrazovky aplikace



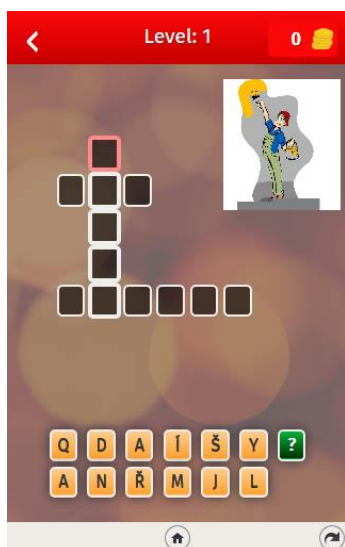
Úvodní obrazovka při prvním spuštění



Úvodní obrazovka s rozehranými hrami



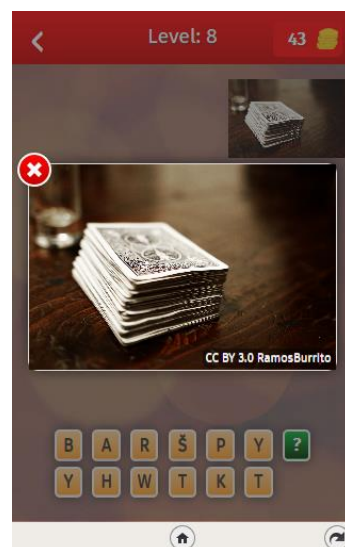
Vytvoření nové hry



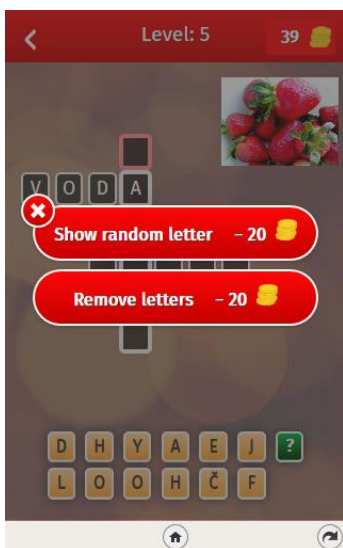
První úroveň hry



Rozehraná hra



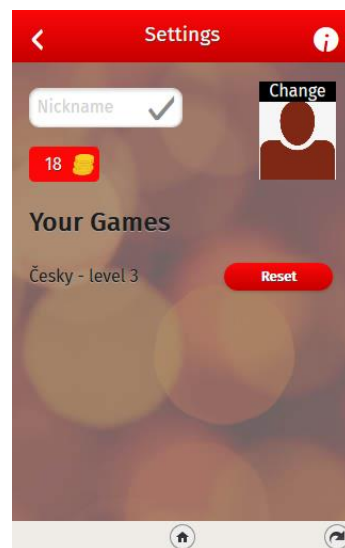
Funkce zvětšení obrázku



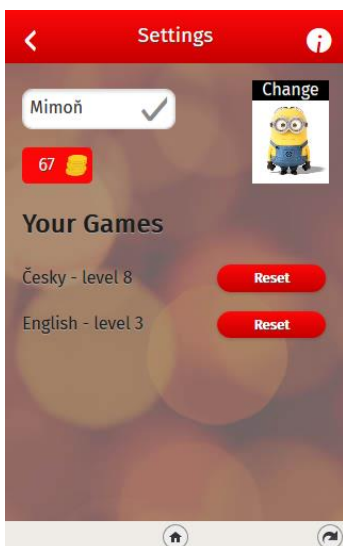
Volba nápovědy



Použitá nápověda „Skrytí písmen“



Obrazovka Nastavení



Nastavení vlastního jména a obrázku



Obrazovka O aplikaci



Fotografie z průběhu testování na mobilním zařízení

## Příloha B: Definice první úrovně hry

Ukázka JSON souboru s definicí první úrovně hry, slov a jejich polohou v mřížce. Struktura souboru vychází z objektu *Word*.

```
1  {
2  "0":{
3      "word": {
4          "name":"malir",      < identifikátor slova
5          "word":[
6              "cs/malir.jpg",  < adresa příslušného obrázku
7              "CC0",          < licence obrázku
8              "m","a","l","i","r"
9          ],
10         "row":2,            < číslo řádku v mřížce, od kterého se bude vypisovat slovo
11         "col":3,          < číslo sloupce v mřížce, od kterého se bude vypisovat slovo
12         "orient":"vertical" < orientace, ve které se bude vypisovat slovo do mřížky
13     },
14     "word1": {
15         "name":"bar",
16         "word":[
17             "cs/bar.jpg",
18             "CC0",
19             "b","a","r"
20         ],
21         "row":3,
22         "col":2,
23         "orient":"horizontal"
24     },
25     "word2": {
26         "name":"bricho",
27         "word":[
28             "cs/bricho.jpg",
29             "public domain",
30             "b","ř","i","c","h","o"
31         ],
32         "row":6,
33         "col":2,
34         "orient":"horizontal"
35     }
36 }
37 }
```

# Příloha C: Zápis z uživatelského testování použitelnosti

## Uživatelské testování aplikace Foxword – Uživatel 1: Lenka

Ovládání plynulé, slova doplňuje rychle, používá gesta z PixWords.

### **Předchozí zkušenosti s PixWords:**

Ano

### **Jak se ti hra líbí?**

*Jo dobrý. Hraje se to dobře.*

### **Přijde Ti ovládání pohodlné/intuitivní?**

*Pohodlné.*

### **Co Ti dělalo problémy nebo co bys změnil/a?**

*Na žádné problémy jsem nenarazila. Akorát mě švalo, že musím cvakat všechna písmenka.*

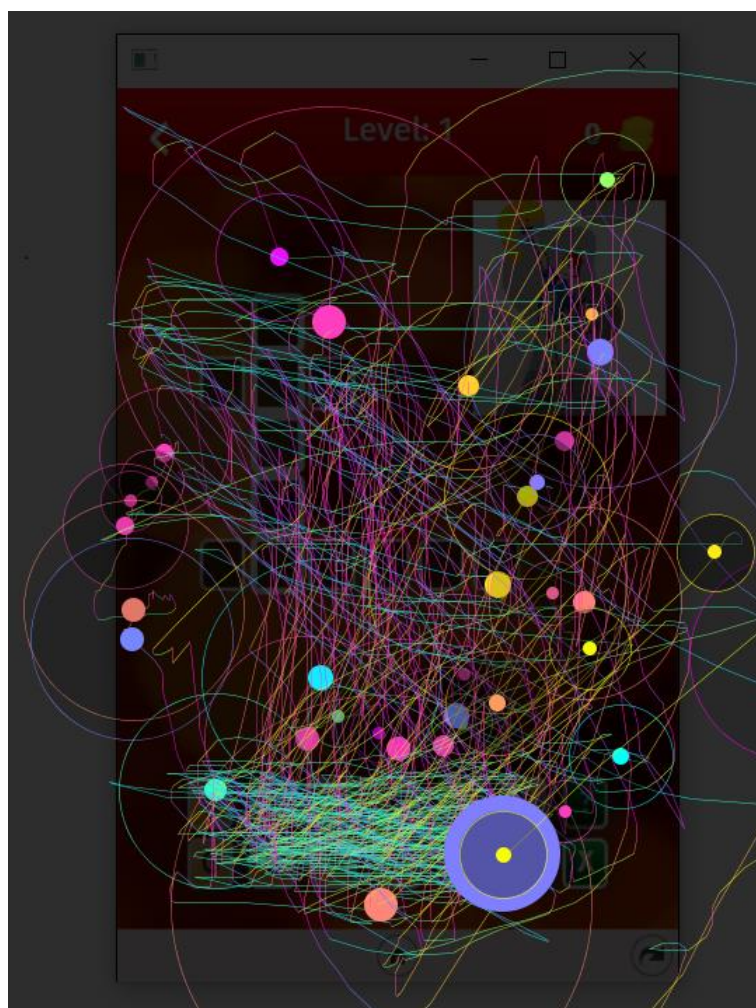
### **Můžeš aplikaci porovnat s hrou PixWords? Je podle Tebe Foxword lepší, stejný nebo horší než PixWords?**

*Je to stejné, až na ty písmenka.*

### **Vadily Ti anglické popisky uživatelského prostředí?**

*Ani jsem si toho nevšimla, až na tu nápovědu.*

## Teplotní mapa pohybu kurzoru – Uživatel 1: Lenka



## **Uživatelské testování aplikace Foxword – Uživatel 2: Blanka**

Ovládání plynulé, slova doplňuje rychle, jedno slovo s nápovědou, používá gesta z PixWords.

### **Předchozí zkušenosti s PixWords:**

Ano

### **Jak se ti hra líbí?**

*Bylo to dobrý, fakt stejný jako PixWords.*

### **Přijde Ti ovládání pohodlné/intuitivní?**

*Ovládalo se to dobře.*

### **Co Ti dělalo problémy nebo co bys změnil/a?**

*Asi nic.*

*Když dáš špatně třetí písmenko a chceš ho vyndat, tak zmizí všechno.*

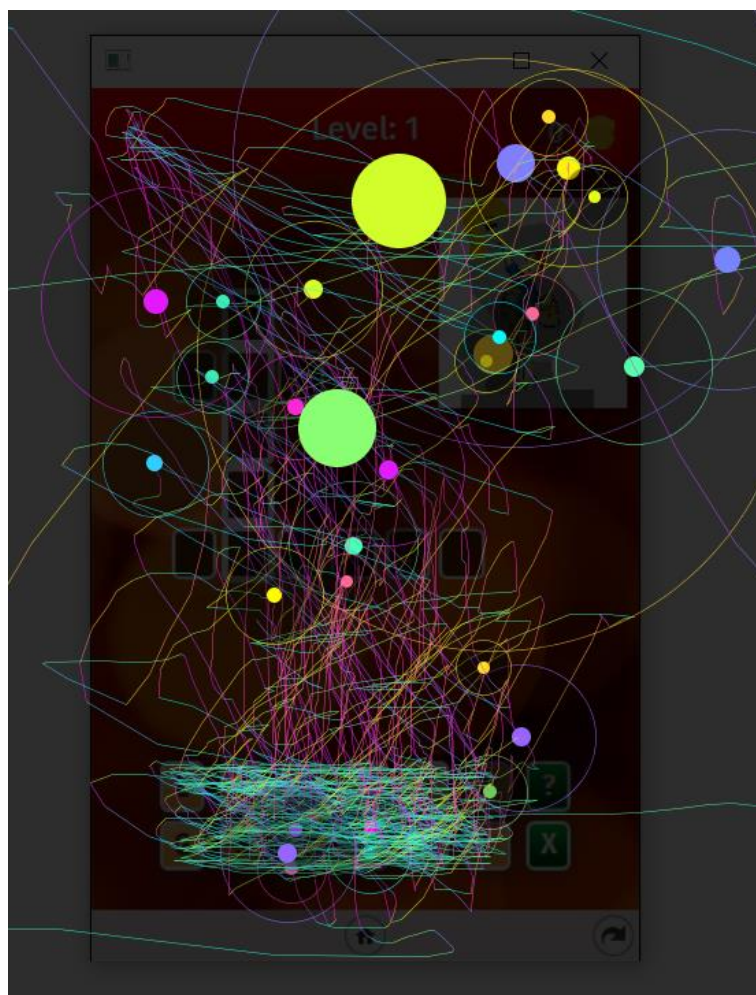
### **Můžeš aplikaci porovnat s hrou PixWords? Je podle Tebe Foxword lepší, stejný nebo horší než PixWords?**

*Je to stejné, líbilo se mi, že se dá změnit obrázek.*

### **Vadily Ti anglické popisky uživatelského prostředí?**

*Ne.*

## Teplotní mapa pohybu kurzoru – Uživatel 2: Blanka



## **Uživatelské testování aplikace Foxword – Uživatel 3: Michal**

Ovládání velice rychlé, slova doplňuje pohotově, používá gesta z PixWords.  
Nejprve změnil v nastavení fotku a jméno.

Na konci testování:

*Když už jsem pochopil ten systém, je to úplně v klidu.  
Tam je nějaké info? Jsem si ani nevšiml.*

### **Předchozí zkušenosti s PixWords:**

Ano

### **Jak se ti hra líbí?**

*Aplikace se mi líbí.*

### **Přijde Ti ovládání pohodlné/intuitivní?**

*Ovládání bylo v pohodě, žádné zádrhely.*

### **Co Ti dělalo problémy nebo co bys změnil/a?**

*Ty písmenka. Bylo by lepší, abych tam nemusel dávat písmenko, když už tam je.*

### **Můžeš aplikaci porovnat s hrou PixWords? Je podle Tebe Foxword lepší, stejný nebo horší než PixWords?**

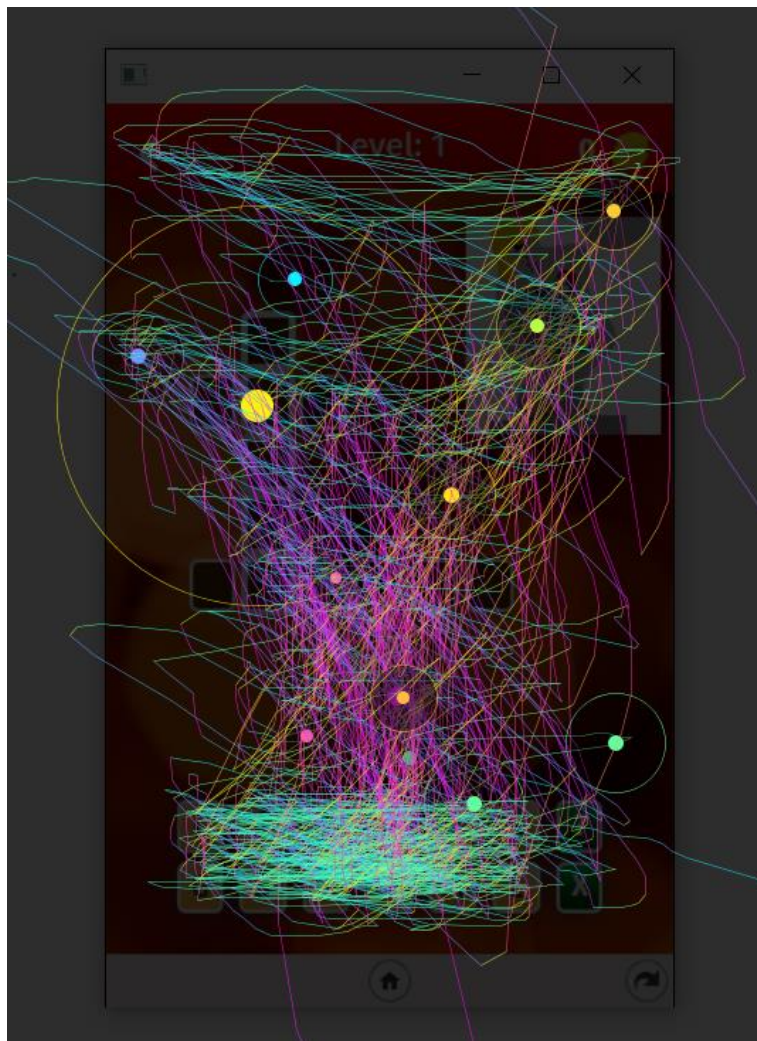
*V PixWords si nemůžu změnit fotku. Kdybys to doladil, tak by to bylo stejný. Kdyby to mělo více profilů pro více hráčů, bylo by to super.*

### **Vadily Ti anglické popisky uživatelského prostředí?**

*Jsem si nevšiml, nevadilo.*



### Teplotní mapa pohybu kurzoru – Uživatel 3: Michal



## **Uživatelské testování aplikace Foxword – Uživatel 4: Vítek**

Ovládání pomalejší, slova doplňuje normálně, několikrát využil nápovědu zobrazení náhodného písmene.

### **Předchozí zkušenosti s PixWords:**

Ne

### **Jak se ti hra líbí?**

*Takový hry mám rád a je s podivem, že jsem si to (PixWords, pozn. autora) ještě nestáhl.*

### **Přijde Ti ovládání pohodlné/intuitivní?**

*Zvykl jsem si na to, je to dobré.*

### **Co Ti dělalo problémy nebo co bys změnil/a?**

*Například ke slovu padat, bych uvítal šipku dolů.*

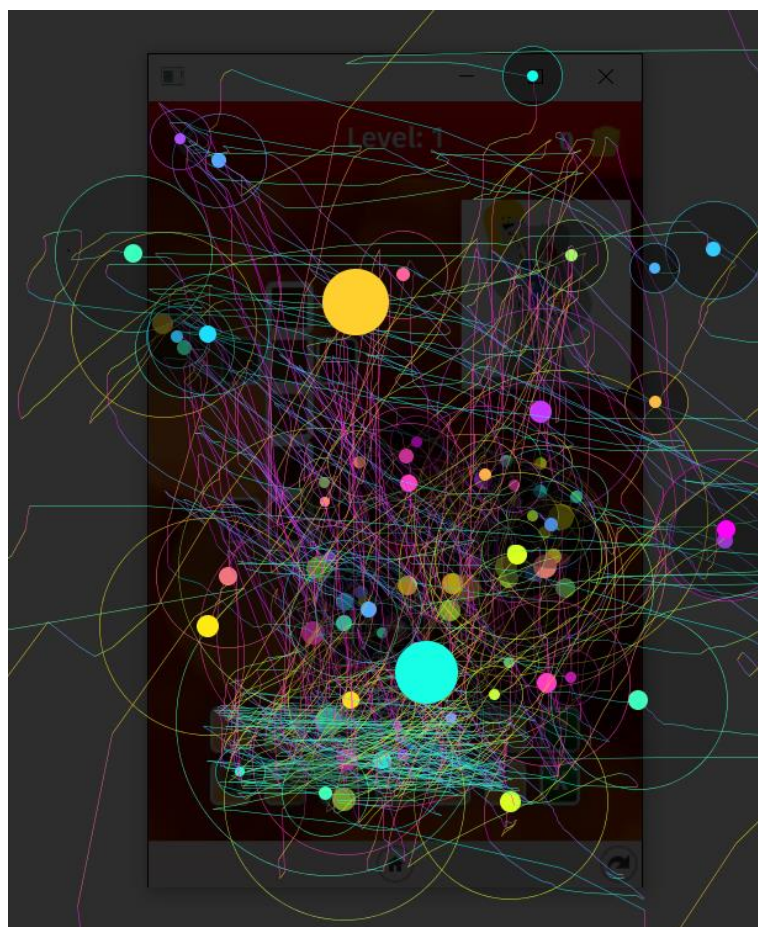
### **Můžeš aplikaci porovnat s hrou PixWords? Je podle Tebe Foxword lepší, stejný nebo horší než PixWords?**

Nemá předchozí zkušenost.

### **Vadily Ti anglické popisky uživatelského prostředí?**

*Nevadily, ani jsem si toho nevšiml. V týchle době je to normální.*

## Teplotní mapa pohybu kurzoru – Uživatel 4: Vítek



## **Uživatelské testování aplikace Foxword – Uživatel 5: Petra**

Ovládání plynulé, slova doplňuje rychle, nápovědy zkoušela, občas se výběr požadovaného slova v křížovce povedl na druhý pokus.

### **Předchozí zkušenosti s PixWords:**

Ne

### **Jak se ti hra líbí?**

*Libila, bavilo mě to.*

### **Přijde Ti ovládání pohodlné/intuitivní?**

*V pohodě.*

### **Co Ti dělalo problémy nebo co bys změnil/a?**

*Když tam bylo doplněné písmeno, nechtěla jsem ho tam dávat znovu.*

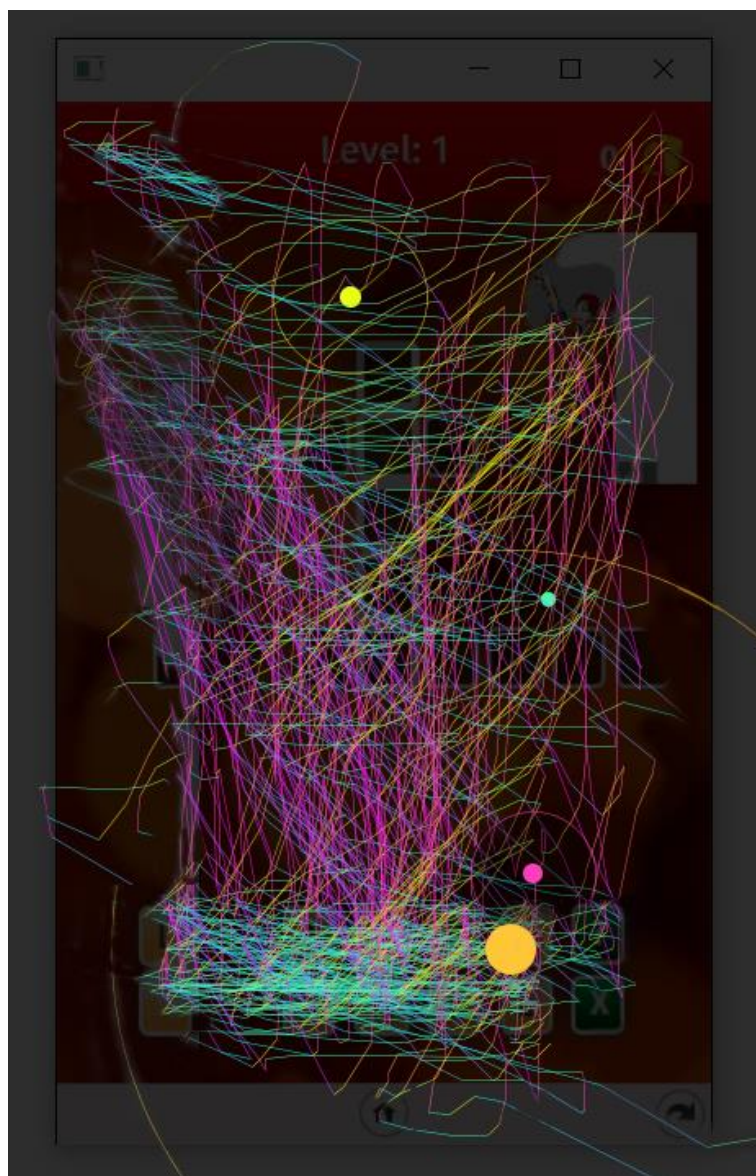
### **Můžeš aplikaci porovnat s hrou PixWords? Je podle Tebe Foxword lepší, stejný nebo horší než PixWords?**

Nemá předchozí zkušenost.

### **Vadily Ti anglické popisky uživatelského prostředí?**

*Vůbec jsem si toho nevšimla.*

## Teplotní mapa pohybu kurzoru – Uživatel 5: Petra



## Příloha D: Disk CD

Součástí bakalářské práce je také přiložené CD obsahující:

- elektronickou verzi bakalářské práce ve formátu PDF;
- zdrojové kódy vytvořené aplikace.

Adresářová struktura CD:

<code>./README.txt</code>	popis adresářové struktury CD
<code>./app</code>	adresář se zdrojovými kódy aplikace
<code>./text</code>	adresář s textem bakalářské práce v PDF