

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Bakalářská práce**

# **Hra Dobble pro mobilní zařízení**

Plzeň 2014

Tomáš Balíček

# Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 9. května 2014

Tomáš Balíček

# Poděkování

Děkuji Ing. Ladislavu Pešíčkovi za odborné vedení mé bakalářské práce a za cenné rady, které mi pomohly tuto práci vytvořit.

V Plzni, dne 9. května 2014

Tomáš Balíček

# Abstract

Mobile applications and games made recently a big step forward. The demand for mobile devices is growing rapidly. Therefore, the development of mobile applications is becoming very interesting. In this area is necessary to examine the issue of game control on different types of mobile devices. Games can be single players or multiplayer. Mobile applications can be made natively for each platform separately or use a multiplatform tool.

This bachelor thesis mainly deals with programming multiplatform games written in HTML and specifically written in PhoneGap framework. The aim of this work is to program Android game with use this framework. It is card game called Dobble. The game consists in finding two identical symbols on two cards. This game is currently available only for Apple iOS (April 2014).



# Abstrakt

Aplikace potažmo hry na mobilních zařízeních prošly v posledních letech velkým rozmachem. Poptávka po mobilních zařízeních rychle roste. A díky tomu se stává vývoj aplikací velice zajímavý. V této oblasti je nutné prozkoumat problematiku ovládání her na různých typech mobilních zařízení, hry mohou být jednoduživatelské nebo pro více uživatelů. Aplikace je možné psát nativně pro každou platformu zvlášť nebo zvolit multiplatformní nástroje.

Tato bakalářská práce se zabývá převážně programováním multiplatformních her psaných v HTML a konkrétně využití frameworku PhoneGap. Cílem této práce je naprogramovat hru pro platformu Android. Jedná se o karetní hru Dobble. Hra spočívá v hledání dvou shodných symbolů na dvou kartičkách. Hra je zatím pouze dostupná v karetní podobě a pro platformu iOS (duben 2014).

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Interakce pro ovládání mobilní hry</b>	<b>2</b>
2.1	Obyčejné telefony . . . . .	2
2.1.1	Ovládání . . . . .	2
2.2	Chytré telefony . . . . .	3
2.2.1	Gyroskop . . . . .	3
2.2.2	Akcelerometr . . . . .	4
2.2.3	Dotykový displej . . . . .	4
2.2.4	Shrnutí . . . . .	6
<b>3</b>	<b>Multiuživatelské hry</b>	<b>7</b>
3.1	Jedno zařízení . . . . .	7
3.2	Komunikace více zařízení . . . . .	7
3.2.1	Bluetooth . . . . .	7
3.2.2	Wi-Fi - Sít'ová komunikace . . . . .	8
<b>4</b>	<b>Pravidla hry</b>	<b>10</b>
4.1	Single player . . . . .	10
4.2	Multi player - jedno zařízení . . . . .	10
4.3	Multi player - více zařízení . . . . .	10
<b>5</b>	<b>Realizace hry</b>	<b>11</b>
5.1	Klient . . . . .	11
5.1.1	Kartičky . . . . .	12
5.1.2	Rozpoznání zvoleného obrázku . . . . .	12
5.1.3	PhoneGap . . . . .	14
5.1.4	Příprava prostředí . . . . .	15
5.1.5	Java . . . . .	18
5.1.6	Javascript . . . . .	19
5.1.7	HTML5 a CSS3 . . . . .	21

---

5.2	Server . . . . .	23
5.3	Komunikační protokol . . . . .	24
5.3.1	Zahájení komunikace serveru . . . . .	24
5.3.2	Zahájení komunikace klienta . . . . .	25
5.3.3	Komunikace mezi serverem a klientem . . . . .	25
5.4	Časově kritické úseky hry . . . . .	26
5.4.1	S ohledem na výkonnost zařízení . . . . .	27
5.4.2	S ohledem na rychlost připojení . . . . .	27
5.4.3	Možná řešení . . . . .	27
<b>6</b>	<b>Uživatelská dokumentace</b>	<b>29</b>
6.1	Single player . . . . .	29
6.2	Multi player - offline . . . . .	29
6.3	Multi player - online . . . . .	30
6.4	Přeložení aplikace a spuštění aplikace . . . . .	31
<b>7</b>	<b>Distribuce hry</b>	<b>32</b>
<b>8</b>	<b>Testování</b>	<b>34</b>
8.1	Ladící konzole Eclipse . . . . .	34
8.2	Ladící nástroje v prohlížeči . . . . .	34
8.3	Emulátor . . . . .	36
8.4	Zařízení . . . . .	37
8.5	Online verze . . . . .	39
8.6	Test s ohledem na rychlost připojení . . . . .	39
<b>9</b>	<b>Závěr</b>	<b>40</b>

# 1 Úvod

Bakalářská práce se zabývá tématem multiplatformních aplikací pro mobilní zařízení. Bude potřeba prozkoumat interakce pro ovládání mobilních her na různých typech telefonů. U telefonů, které nám nabízí širší možnosti využití nalézt, které z jejich vyspělých funkcí by se dali použít pro ovládání her. U multiuživatelského režimu bude potřeba vhodné nalezení a propojení zařízení, které spolu budou komunikovat.

V realizaci hry si projdeme konkrétní framework použitý na vývoj hry, která je tvořena převážně pomocí webových technologií a popíšeme si převod do nativní aplikace. Díky tomu získáme aplikace, které budou přenosné na různé mobilní platformy, budou skvěle vypadat a navíc budou schopné využívat specifické vlastnosti konkrétních zařízení. U režimu online bude probrána problematika komunikačního serveru a popis komunikace mezi jednotlivými zařízeními a vhodné nalezení spoluhráčů.

Po vytvoření aplikace si ukážeme, jak aplikaci vhodně testovat v ladící konzoli a využívat ladící nástroje moderních prohlížečů. Poté bude aplikace nasazena do emulátoru a v posledním kroku bude aplikace nasazena přímo do fyzického zařízení pro otestování různých rozměrů displeje a hardwarových požadavků aplikace.

Posledním krokem v životním cyklu aplikace je distribuce aplikace do příslušného internetového obchodu. Ukážeme konkrétní příklad nasazení naší hry do Android Store společnosti Google.

## 2 Interakce pro ovládání mobilní hry

Možnosti ovládání mobilních her můžeme rozdělit podle druhu mobilního zařízení:

### 2.1 Obyčejné telefony

Jedná se o kategorii telefonů, které jsou vybaveny hardwarovou klávesnicí ve formě číslic 1-9 a několika dalšími kontextovými tlačítky. Tyto zařízení jsou také označována jako hloupá zařízení nebo anglicky *cell phone*. Na zařízení tohoto typu není možné instalovat další aplikace. Jedná se o zařízení s minimálními funkcemi. Zařízení slouží ve většině případů pouze k telefonování, psaní SMS, popřípadě MMS a novější typy těchto telefonů byly vybaveny i kamerou. I když se může zdát, že tato zařízení jsou dávnou historií i v dnešní době najdou určité využití. Hlavní výhodou je jejich výdrž, kdy bez připojení do nabíječky vydrží nabitě dlouhou dobu ve srovnání s chytrými telefony. Jejich další výhodou je bytelnost, při pádu z výšky na zem většinou nedochází k takovému poškození jako u chytrých zařízení, jedno z nejmohutnějších míst bývá displej a ten je většinou menších rozměrů a šasi telefonu z plastu. A nevýhody těchto zařízení mohou být pro určitou skupinu lidí naopak výhodou, protože tyto telefony nejsou například vybaveny ani GPS senzory, takže doinstalovat aplikaci pro špehování zařízení pomocí GPS senzoru může být poněkud složité. V nabídce od operátorů je to například telefon Samsung E1200M Keystone 2, u kterého se cena pohybuje kolem 300,- Kč za kus.

#### 2.1.1 Ovládání

Pro tuto platformu jsou možnosti ovládání omezené. Tento druh telefonů se ovládá pomocí hardwarové klávesnice, která má klasické rozvržení tlačítek 0 až 9 nebo u novějších zařízení byla již QWERTY klávesnice. Pokud by byla hra naprogramována pro tento druh telefonů, každý obrázek by musel mít své číselné označení a na hardwarové klávesnici by se tak jednoduše stisklo dané tlačítko, které by odpovídalo obrázku. Ale vzhledem k tomu, že i propozice samotného displeje nejsou na vysoké úrovni bylo by to prakticky nereálné.

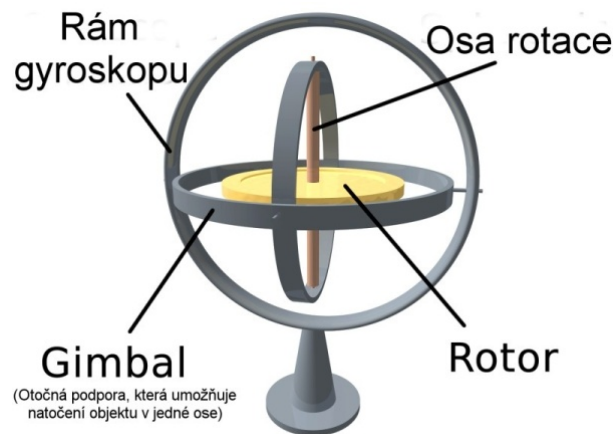
## 2.2 Chytré telefony

Chytré telefony anglicky *smartphone* jsou zařízení, která jsou vybavena pokročilým mobilním operačním systémem a aplikační rozhraní nám umožňuje instalaci a vytváření nových programů. Jedná se například o operační systémy: iOS, Android, Windows Phone, Firefox OS, BlackBerry OS a další. V této kategorii mobilních telefonů jsou možnosti ovládání již pestřejší. Chytré telefony mají spoustu výhod jako například: doinstalování velkého množství aplikací, které jsou většinou ke stažení zdarma popřípadě za malý poplatek, jsou vybaveny velkým množstvím senzorů a hlavně dotykovým displejem. Pro ovládání her se často využívá gyroskop. Jedná se o senzor který detekuje pohyb i v třetí ose. Chytré telefony nám ale přináší také spousty nevýhod naproti obyčejným telefonům: jednou z hlavních nevýhod je výdrž baterie, zatímco obyčejné telefony vydrží bez nabíjení i týden u chytrých telefonů je většinou potřeba nabíjet každý den, ale to je také dáno funkcemi telefonu, který již není využíván jen pro telefonování a psaní SMS. Pak jsou to určité rozměry, zatímco do příchodu chytrých telefonů bylo snahou většiny výrobců co nejvíce snížit hmotnost a rozměry telefonu dnes je to právě naopak a většina výrobců se přetahuje o to kdo bude mít větší telefon potažmo displej. A asi největší nebezpečí chytrých telefonů jsou škodlivé aplikace. Viry jsou spojeny s hlavní výhodou chytrých telefonů a to s instalací vlastních aplikací. S postupným nárůstem těchto zařízení bude riziko stoupat. Nyní jsou aplikace nahrávané do mobilních obchodů velmi striktně kontrolovány, takže k infikaci zařízení dochází velice zřídka. Pokud se podíváme do nabídky operátorů převažují právě chytrá zařízení. Z těch nejlepších, které nyní můžeme na trhu nalézt je to Samsung Galaxy S5 nebo Apple iPhone 5, kde cena těchto zařízení je podobná jako u lepšího notebooku.

### 2.2.1 Gyroskop

Gyroskop [14] nám určí v jaké poloze se telefon nachází a pohyb který s telefonem provádíme. První kdo použil gyroskop v mobilním zařízení byl Apple ve svém modelu iPhone 4 v roce 2010. Gyroskop je vlastně setrvačnický tvořený těžkým kolem otáčejícím se v ložiscích s nepatrným třením, který si po roztočení zachovává polohu osy své rotace v prostoru a dokáže tak určit svou orientaci. Většinou je setrvačnický uložen v takzvaných Cardanových závěsech. Rozmach těchto zařízení nastal v 70. letech kdy se začali používat v balistických raketách nebo torpédách a například také u letadel nebo navigace

lodí. V mobilních hrách má gyroskop také velké využití. Používá se například pro ovládání her automobilových závodů, kdy simulujeme natočením telefonu směr vlevo nebo vpravo. Další hry, kde je gyroskop využíván jsou například hry, kdy se snažíme dostat kuličku do cíle přes bludiště na displeji. Gyroskop nám pomůže k tomu, že simulace na displeji je fyzikálně odpovídající skutečnosti, kdy bychom měli skutečnou destičku a na té kutáleli kuličku.



Obrázek 2.1: Znázornění gyroskopu [14]

### 2.2.2 Akcelerometr

Akcelerometr je zařízení, které slouží ke zjištění zrychlení při pohybu telefonu. A může také měřit orientaci telefonu. To můžeme využít ve hrách při různých gestech. Nebo například v hudebních přehrávačích, kdy se při zatřesení přepne na další skladbu. Velké využití má akcelerometr například v krokoměrech, kdy je většina moderních telefonů schopna pomocí nějaké aplikace zaznamenat délku ušlé trasy.

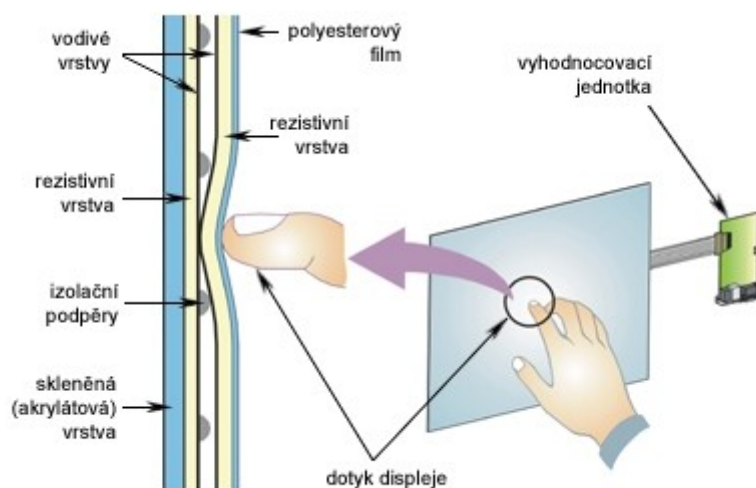
### 2.2.3 Dotykový displej

Dotykové displeje můžeme dělit na dvě hlavní kategorie, které se používají u mobilních zařízení a to: rezistivní a kapacitní. Vzhledem k tomu, že v dnešní době se pomocí dotykových displejů ovládá téměř většina mobilních zařízení jejich vývoj pokročil v posledních letech velice kupředu. Velký boom

dotykových displejů nastal v roce 2007 kdy Apple představil svůj iPhone. Ale nebyl to zdaleka první telefon s dotykovým displejem toto prvenství si drží Simon PDA cell phone, který v roce 1993 představila společnost BellSouth a IBM.

### Rezistivní displeje

Rezistivní neboli odporové displeje [12] můžeme ovládat prakticky čímkoli. V okamžiku kdy se displeje dotkneme, horní dotyková vrstva se prohne a spojí se se spodní elektrovodivou vrstvou čímž mezi nimi začne procházet elektrický proud. Řídící a vyhodnocovací modul následně vyhodnotí polohu a velikost bodu dotyku. U těchto displejů se tedy velice špatně provádí gesta a horní vrstva musí být vyrobena z měkkého a pružného materiálu, tudíž často trpí poškrábáním a proto tento druh displejů nahradily displeje kapacitní.



Obrázek 2.2: Princip rezistivního displeje [12]

### Kapacitní displeje

Kapacitní displeje [12] v dnešní době ovládly trh s mobilními zařízeními. Jejich funkčnost spočívá ve vodivosti lidského těla. Jakmile se dotkneme displeje, naruší se elektrostatické pole displeje a mezi jeho povrchem a špičkou



prstu vznikne kapacita, která uzavře elektrický obvod. Výhody kapacitních displejů tedy jsou, vysoké rozlišení displejů, vysoká světelná propustnost a dotyková plocha může být vyrobena z vysoce pevných a odolných skel. Nevýhodou je, že displeje reagují pouze na dotyk prstu nebo jiného vodivého předmětu, takže například na dotek v rukavici displej nezareaguje. Ale někteří výrobci doplňují kapacitní displej o další super-citlivou vrstvu, díky ní není problém ovládání například nehty nebo v rukavicích. Například jsou to Lumie od finského výrobce Nokia.

#### **2.2.4 Shrnutí**

Dnešní mobilní zařízení potažmo chytré telefony umožňují využití velkého množství senzorů a dalších příslušenství. Pro naši hru však zůstaneme u využití dotykového displeje a připojení k síti. Dotykový displej nám umožní vše co pro ovládání hry potřebujeme v multiuživatelské verzi i možnost přístupu dvou hráčů, kde každý bude využívat svoji polovinu displeje. Síťové připojení nám umožní realizovat multiuživatelskou hru online, kde bude mít každý z hráčů své zařízení a tyto zařízení spolu budou komunikovat přes síť.

## 3 Multiuživatelské hry

Multiuživatelské hry na mobilních zařízeních můžeme rozdělit na hry, které se hrají na jednom zařízení a hráči se o jedno zařízení střídají nebo hry pro dva a více hráčů, kde každý hraje na své polovině zařízení. Dalším typem jsou hry, kde každý z hráčů již hraje na svém zařízení a tato zařízení spolu musí na dálku komunikovat.

### 3.1 Jedno zařízení

Realizace této multiuživatelské hry je velice jednoduchá, jedná se o takzvanou offline hru, kdy není nutné navazovat například síťové spojení, pokud hra nemá další nadstandardní možnosti. Hráči se mohou o jedno zařízení střídát například po určitém čase nebo vždy po odehraném kole. Ve hře Dobble jsem se rozhodl jako jeden z herních módů implementovat možnost, hrát na jednom zařízení současně. Hra je určena pro 2 hráče, z nichž každý má k dispozici jednu polovinu displeje.

### 3.2 Komunikace více zařízení

Pokud jsou ve hře zapojeny dvě a více zařízení, realizace této hry se značně zkomplikuje. Již musí být k dispozici nějaké spojení pomocí kterého budou dvě daná zařízení komunikovat mezi sebou. Jedná se o takzvanou online verzi. Máme na výběr mnoho typů spojení, která můžeme mezi zařízeními navázat.

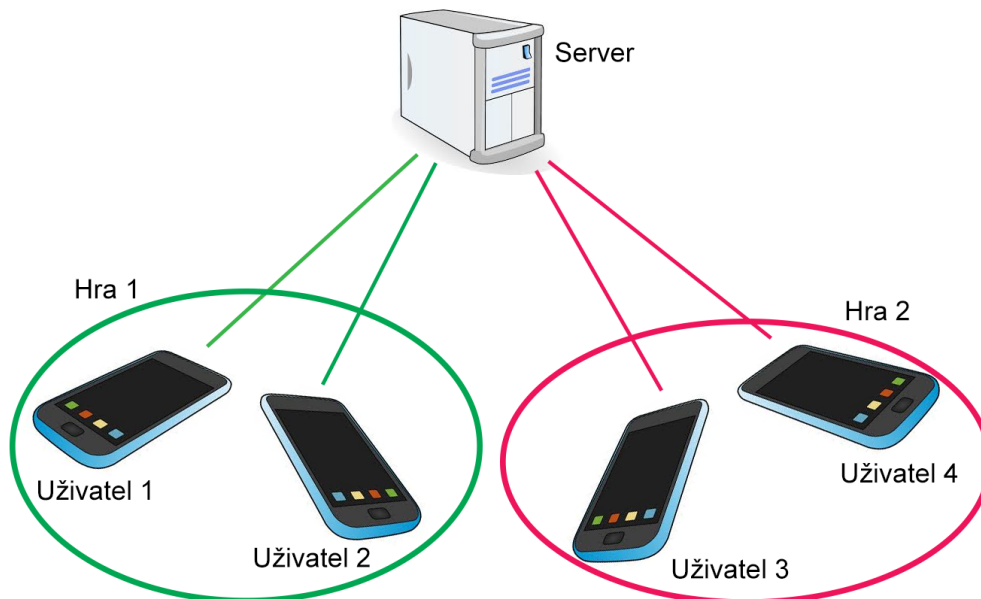
#### 3.2.1 Bluetooth

Bluetooth [15] byl vytvořen firmou Ericsson v roce 1994. V dnešní době ho nalezneme téměř v každém chytrém telefonu. Tato technologie je definovaná standardem IEEE 802.15.1 a spadá do kategorie PAN, tzv. osobních počítačových sítí a to nám přináší velká omezení. Aby mohli dva spoluhráči vytvořit spojení, musí se nacházet v blízkém okolí nejlépe jednoho pokoje. Udávaný dosah je 100 metrů u Bluetooth 4.0 které je nasazováno do nejnovějších mo-

bilních zařízení. Tato technologie bývá pro multiuživatelské hry využívána velice zřídka.

### 3.2.2 Wi-Fi - Síťová komunikace

Wi-Fi je technologie, bez které se v dnešní době neobejde žádný chytrý telefon. Slouží pro bezdrátovou komunikaci v počítačových sítích. Síťová komunikace nám již přináší mnohem větší možnosti. Zařízení již nejsou omezeny vzdáleností, ve které se od sebe nachází a nemusí spolu komunikovat na přímo, ale přes nějaký komunikační server. Tuto možnost využívám i v herním multiuživatelském online režimu. Hráči se připojují k serveru a komunikují viz obrázek 3.1. Má to samozřejmě i své nevýhody, každé zařízení musí mít připojení k síti popřípadě k internetu a server musí být spuštěný. Pokud tedy z nějakého důvodu server selže, není možné ve hře pokračovat ani vytvořit hru novou.



Obrázek 3.1: Komunikace zařízení přes server

**Nalezení spoluhráče**

Nalezení je možné řešit více různými způsoby. Například ve hře Dobble je omezení, že v každé hře mohou být současně maximálně 2 spoluhráči. Po připojení hráče k serveru je mu nabídnuto jestli chce vytvořit novou hru nebo se připojit do nějaké již vytvořené. Pokud si uživatel vybere připojení do již vytvořené hry jsou tu dvě možnosti. Zaprvé mu nabídnout všechny hry, do kterých je možno se momentálně se připojit nebo zadat nějaké ID, které jasně identifikuje danou hru. Pokud je tedy ve hře požadovaný počet spoluhráčů, hra může začít.

## 4 Pravidla hry

Hra Dobble spočívá v hledání stejných obrázků na dvou různých kartičkách. Přičemž obrázky na každé z kartiček jsou různě velké. Po nalezení stejného obrázku se na daný obrázek klikne a hra vyhodnotí jestli se obrázek opravdu shoduje. Nyní jsou 3 různé herní režimy:

### 4.1 Single player

V tomto režimu se hráč snaží nasbírat co nejvyšší skóre, dokud mu nedojdou životy. Na začátku hry má hráč 3 životy. Hra je rozdělena na jednotlivá kola a v každém následujícím kole se sníží čas na uhodnutí stejného obrázku. V každém kole je potřeba nasbírat 5 správných uhodnutí, aby se postoupilo do dalšího kola. Při postupu do dalšího kola je hráči přidán jeden nový život. Pokud hráč vybere špatný obrázek nebo mu dojde čas na nalezení obrázku je mu odebrán jeden život. Hra je ukončena po odebrání všech životů.

### 4.2 Multi player - jedno zařízení

V tomto režimu mohou hrát dva hráči proti sobě. Každému hráči patří jedna ze dvou kartiček. Před zahájením hry se musí nastavit počet bodů, který je nutný pro dosažení vítězství plus si každý hráč může zvolit barvu své kartičky a zadat své jméno. Opět při zvolení správného obrázku je danému hráči přičten jeden bod a při špatném pokusu jeden bod odečten.

### 4.3 Multi player - více zařízení

V tomto režimu se hraje na dvou různých zařízeních po síti. Před začátkem hry se opět zvolí kolik bodů je nutné pro celkové vítězství. Po dosažení zvolených bodů se hra ukončí a každému hráči je oznámeno jestli vyhrál nebo naopak prohrál. Poté se hra může opakovat.

## 5 Realizace hry

Na realizaci celé hry bylo použito velké množství technologií. Server je napsán v nízkourovňovém jazyce C. Nativní jazyk Androidu je Java a proto rozšíření, která zde potřebujeme pro síťovou komunikaci jsou napsána v Javě a o obsluhu WebView tedy o GUI celé aplikace se stará značkovací jazyk HTML5 s kaskádovými styly CSS3 a jsou obsluhovány multiplatformním, objektově orientovaným skriptovacím jazykem Javascript. Výsledná práce je vidět na obrázku 5.1.



Obrázek 5.1: Hra Dobble v1.0

### 5.1 Klient

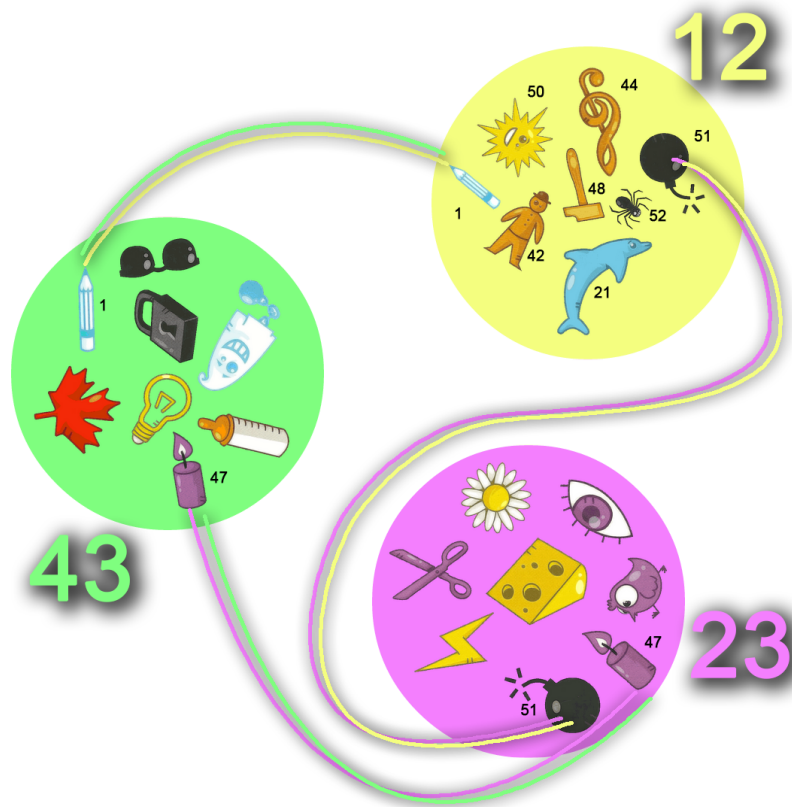
Pro vývoj mobilní aplikace jsem použil framework PhoneGap [6], který umožňuje napsat aplikaci pomocí HTML5, Javascriptu a CSS3. Velká výhoda tohoto frameworku je, že aplikace bude jen s malými úpravami multiplatformní.

### 5.1.1 Kartičky

Ve hře je celkem 55 různých kartiček a na každé z nich je 8 obrázků. Každé dvě kartičky mají jeden obrázek společný, ale obrázek nemusí být stejně velký a navíc různě natočený viz obrázek 5.2. Kartičky byly převzaty z papírové podoby hry Dobble. Znamená to, že jednotlivé kartičky byly jedna po druhé naskenovány a poté upravovány v programu Photoshop do požadované velikosti, podoby a průhledného pozadí. Průhledné pozadí je požadováno proto, aby si hráč mohl zvolit libovolnou barvu. Na kartičkách je celkem 57 různých obrázků, které se různě střídají, aby byla dodržena logika hry a opravdu každá kartička měla s každou právě jeden shodný obrázek. Musel jsem tedy vypracovat matici o velikosti 55 x 55, kde je zaznamenán shodný obrázek každých dvou kartiček. Pokud tedy uživatel klikne na nějaký z obrázků, my víme čísla kartiček, mezi kterými hledá shodný obrázek a číslo daného obrázku, které zjistíme z mapy každé kartičky viz dále. Poté se tedy můžeme podívat do naší matice, kde řádek matice bude číslo jedné kartičky a sloupec číslo druhé kartičky a pokud hodnota na tomto místě bude rovna číslu zvoleného obrázku, uživatel zvolil správně. Pokud se hodnota bude lišit, hned víme, že uživatel zvolil špatný obrázek a můžeme zobrazit chybovou hlášku.

### 5.1.2 Rozpoznání zvoleného obrázku

Jak již víme po stisknutí displeje potřebujeme identifikovat, na který z obrázků bylo kliknuto. Každá kartička má tedy vygenerované polygony a každý polygon odpovídá jednomu obrázku na kartičce, který chceme zvolit. Při doteku displeje si tedy zjistíme přesné souřadnice kam bylo kliknuto. Jelikož je hra přenosná na více zařízení, musí se řešit problém s různým rozlišením displejů. Pokud klikneme uprostřed displeje na zařízeních z různým rozlišením dostaneme pokaždé jiné souřadnice. Souřadnice polygonů jsou na kartičkách o velikosti 1000px x 1000px pokud je tedy šířka displeje také 1000px nemusí se přepočítávat nic. Pokud je například pouze 500px je tedy o polovinu menší tím pádem vynásobíme dvakrát x souřadnici bodu, kde se uživatel dotkl. Nyní víme kam přesně uživatel klikl a můžeme, tak zjistit o jaké číslo obrázku se jedná. Proto má každá kartička vygenerovanou souřadnicovou mapu polygonů a každý polygon má přiřazeno číslo od 1 do 57, jak bylo zmíněno výše 57 je počet všech obrázků, které se v této hře nacházejí. Pokud se tedy bod, kam uživatel klikl nachází v nějakém z polygonů kartičky, víme tedy jaké číslo obrázku to je a nic nám nebrání v tom zjistit, jestli se uživatel mýlí nebo se trefil a zvolil shodný obrázek na zobrazených kartičkách.

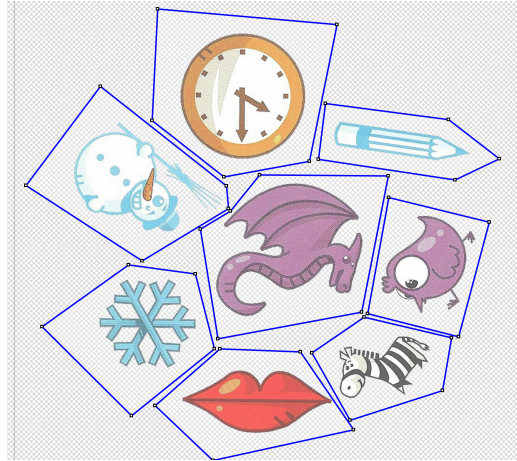


Obrázek 5.2: Ukázka shodných obrázků na kartičkách

### Generování polygonů

Na generování těchto polygonů používám online nástroj Summer [10]. Do tohoto online nástroje si nahrajeme obrázek, který nám bude sloužit jako podklad. Po úspěšném nahrání obrázku již můžeme začít generovat polygony na zvoleném pozadí. Přepneme se tedy do možnosti generovat polygony a začneme generovat jednotlivé polygony viz obrázek 5.3. Pokud máme zvoleny všechny polygony na dané kartičce, můžeme z tohoto náhledu vygenerovat html kód, který je klasická HTML mapa viz obrázek 5.4. V tagu *map* jsou tagy *area*, kde v atributu *coords* nalezneme souřadnice námi požadovaných polygonů.





Obrázek 5.3: Generování polygonů jedné kartičky

```


<map name="map">
  <area shape="poly" coords="225, 681, 205, 874, 390, 857, 414, 719" />
  <area shape="poly" coords="503, 292, 474, 491, 574, 487, 602, 279" />
  <area shape="poly" coords="314, 519, 307, 678, 461, 683, 509, 551, 376, 489, 317, 516" />
  <area shape="poly" coords="216, 392, 60, 435, 79, 670, 286, 624, 296, 484" />
  <area shape="poly" coords="242, 276, 412, 283, 486, 288, 431, 497, 212, 396" />
  <area shape="poly" coords="88, 182, 269, 235, 220, 290, 130, 398, 26, 245" />
  <area shape="poly" coords="436, 144, 357, 202, 384, 256, 493, 264, 608, 207, 548, 99" />
  <area shape="poly" coords="145, 70, 296, 56, 327, 110, 303, 170, 168, 162" />
</map>

```

Obrázek 5.4: Vygenerovaná HTML mapa

### 5.1.3 PhoneGap

PhoneGap je webová stránka zabalená jako nativní aplikace, byl navržen firmou Nitobi a v roce 2011 zakoupen firmou Adobe Systems. V současné době se těší velké popularitě a je použit pro velké množství mobilních aplikací.

Nativní aplikace pro jednotlivé platformy budou vždy pravděpodobně lepší z hlediska rychlosti a vzhledu odpovídajícího dané platformě. Ale vzhledem k tomu, že aplikace pro jednotlivé platformy je nutné vyvíjet každou zvlášť, bude to vždy velice nákladné řešení.

Prvním krokem je tedy napsat webovou aplikaci v HTML5, CSS, JS a po-

kud chceme, aby se aplikace podobala co nejvíce nativní, tak je dobré použít framework, který má již předpřipravené komponenty s co nejvíce nativním vzhledem například jQuery mobile. Za druhé je potřeba zabalení celé webové aplikace do nativní pomoci PhoneGap. Tu si lze představit, jako předpřipravené nativní aplikace pro jednotlivé platformy, které mají přes celou plochu webový prohlížeč, kam se načte právě webová aplikace. Hlavní výhodou oproti běžnému prohlížeči je to, že můžeme danou aplikaci odeslat do App Store, Google Play nebo WP Store a také máme přístup ke všem systémovým API jako třeba geolokaci, fotoaparátu, kontaktům nebo filesystému.

Další výhodou je velké množství pluginů, které pro vývoj můžeme použít a pokud nám nějaký schází, je možné si ho pro danou platformu v nativním jazyce napsat. Na obrázku 5.5 je vidět, že na všech nejrozšířenějších platformách máme již implementován přístup ke všem běžným vlastnostem, které mobilní zařízení nabízejí.

PhoneGap je nyní možné použít na všechny momentálně známé operační systémy chytrých telefonů. Od Amazon Fire OS, Firefox OS, Ubuntu až po iOS, Android, Windows Phone potažmo Windows 8.

Psát tedy aplikace jako webové se vyplatí. Jen pokud by byla aplikace náročnější a jednalo se o akční hru, asi je lepší přemýšlet o nativní aplikaci. Ale vzhledem k tomu, že vývoj webových prohlížečů a webových technologií jde velkým tempem kupředu, určitě je to dobrá volba do budoucna. Již pomalu začínají některé browsery podporovat potřebná API pro přístup k vlastnostem mobilních zařízení.

#### 5.1.4 Příprava prostředí

Abychom mohli naši aplikaci začít vyvíjet je zapotřebí mít nainstalované prostředí Eclipse, které je volně ke stažení na stránkách <http://eclipse.org>. Pro vývoj naší aplikace stáhneme verzi Eclipse pro Java vývojáře, kterou nainstalujeme klasickým způsobem. Pro vývoj mobilních aplikací je potřeba doinstalovat Android SDK a ADT. Díky tomuto pluginu získáme kompletní IDE pro vývoj Android aplikací. To znamená, že získáme i Android emulátor, ve kterém budeme moci aplikaci spouštět a testovat.

	iPhone / iPhone 3G	iPhone 3GS and newer	Android	Blackberry OS 6.0+	Blackberry 10	WebOS	Windows Phone 7 + 8	Symbian	Bada
Accelerometer	✓	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓	✓
Compass	X	✓	✓	X	✓	✓	✓	X	✓
Contacts	✓	✓	✓	✓	✓	X	✓	✓	✓
File	✓	✓	✓	✓	✓	X	✓	X	X
Geolocation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Media	✓	✓	✓	X	✓	X	✓	X	X
Network	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Alert)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Sound)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Vibration)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Storage	✓	✓	✓	✓	✓	✓	✓	X	X

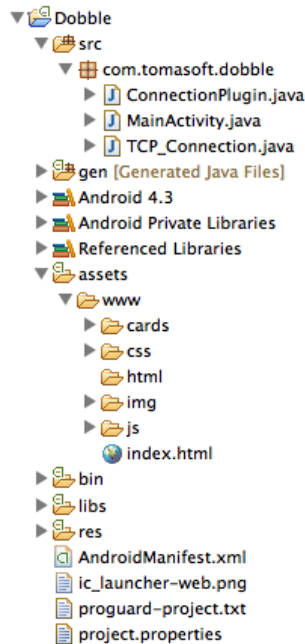
Obrázek 5.5: Tabulka podporovaných vlastností jednotlivých platform

## Založení prvního projektu

Před tím než založíme nový projekt si ještě musíme stáhnout PhoneGap plugin, který nalezneme na <http://phonegap.com/download>, kde si můžeme zvolit v závislosti na operačním systému jaký balíček chceme. Stažený soubor si po stažení rozbálíme na disku. Založení nového projektu probíhá stejně jako u standardní nativní Android aplikace, zvolíme tedy z menu *New Project* a ze zobrazené nabídky vybereme *Android* -> *android Application Project* poté si zadáme jméno naší aplikace, minimální verzi API, na kterém bude možné naši aplikaci spustit a přejdeme na stránku, kde si zvolíme ikonku naší aplikace. V posledním kroku si vytvoříme Prázdnou *MainActivity* a můžeme pokračovat s přidáním frameworku PhoneGap. Celá struktura aplikace je vidět na obrázku 5.6. V adresáři *assets/www/* jsou kompletní zdrojové kódy webové aplikace s přidanou JavaScriptovou knihovnou PhoneGap, kterou bychom pro spuštění na standardním webu nepotřebovali. Hlavním souborem je samozřejmě *index.html*, který je poté zavolán v hlavní aktivitě aplikace viz kapitola 5.1.5.

## Úprava souboru AndroidManifest.xml

Aby naše aplikace mohla přistupovat například k internetu, je potřeba nejdříve definovat přístup v rámci souboru *AndroidManifest.xml*. Možnosti nastavení jsou poměrně široké, tak si ukážeme alespoň základní, která jsem použil v



Obrázek 5.6: Struktura Android aplikace s přidáním frameworkem Phone-Gap

aplikaci Dobble nebo taková, která by mohla být alespoň využita. Základním oprávněním, které potřebujeme pro hraní verze multiplayer online je oprávnění **INTERNET**. Pokud zapomeneme přidat řádek s tímto oprávněním, budeme marně zkoumat, proč se naší aplikaci nedaří připojit k síti. Další oprávnění, které je v aplikaci využíváno je **VIBRATE**. Při docházejícím čase v režimu single player je uživatel upozorněn v posledních 3 sekundách prodlužujícími se vibracemi. V budoucnosti by se mohlo hodit například oprávnění k získání polohy uživatele z toho důvodu, abychom mohli v režimu multiplayer online nabídnout hráče z nejbližšího okolí. Možnosti geolokace jsou dvě a to **ACCESS\_COARSE\_LOCATION** a **ACCESS\_FINE\_LOCATION** na první pohled můžou být zaměňovány, ale je dobré přesně vědět, která se na co hodí. Rozdíly mezi těmito oprávněními je ten, že získání přesné polohy GPS může být velice časově náročné, pokud jsme například v zabydlené oblasti nebo přímo v budově, kde není přímý výhled na oblohu. Proto lze použít první zmiňovaný způsob geolokace **ACCESS\_COARSE\_LOCATION**, na základě síťové geolokace. Tento způsob je zpravidla rychlejší, ale méně přesný a je založený na získávání polohy z pozemních vysílačů mobilních operátorů a Wi-Fi sítí. Druhý způsob **ACCESS\_FINE\_LOCATION** umožňuje při-

stupovat k přesné poloze založené na GPS čipu v zařízení.

### 5.1.5 Java

Výkonný kód klienta je napsán v jazyce Java ve vývojovém prostředí Eclipse s pluginem Android SDK Manager. Přijímání a odesílání zpráv serveru se zajišťuje právě zde. O funkčnost se starají 4 třídy.

#### Třída *MainActivity*

Výchozí třída celého frameworku PhoneGap, která se stará o vytvoření webového okna pro zobrazení aplikace. Hlavní a jediná metoda této třídy je `onCreate()`, která je vidět na obrázku 5.7. Této metodě zadáme cestu k souboru, ve kterém se nachází hlavní soubor webové aplikace.

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    super.loadUrl("file:///android_asset/www/index.html");
}
```

Obrázek 5.7: Metoda `onCreate()`, která je srdcem celé aplikace

#### Třída *ConnectionPlugin*

`ConnectionPlugin` je třída, která zachytává požadavky, které přichází z webového rozhraní a jsou určeny pro komunikaci se serverem. V této třídě se nachází dvě metody `execute()` a `callJavaScript()`. Metoda `execute()` zachytává požadavky z `WebView` a dále je zpracovává a druhá metoda je určena pro komunikaci z nativního prostředí do `WebView`. Pokud je tedy zaslán v multiuživatelské hře online požadavek na vytvoření nové hry, je nejprve zaslán požadavek z `WebView`, kde ho metoda `execute()` zpracuje a odešle potřebné informace na server, který po zpracování a vytvoření nové hry vrátí ID hry. A metoda `callJavaScript()` zašle daný výsledek do `WebView`, kde je o tom informován daný uživatel.

### Třída *TCP\_Connection* a *ListenThread*

Tyto dvě třídy jsou umístěny v souboru `TCP_Connection.java` a mají za úkol vytvoření a udržování komunikace se serverem. Po vytvoření nové instance se v konstruktoru třídy `TCP_Connection` vytvoří nové vlákno třídy `ListenThread`, která se stará o příjem paketů ze serveru a předává je dále na `WebView`, kde přijatá data dále zpracovává Javascript.

#### 5.1.6 Javascript

`WebView` klienta bylo napsáno v prostředí programu `PHPStorm`. Pro lepší práci s Javaskriptem [3] je použita knihovna `jQuery` [7] a `jQuery mobile` [8]. Dále bylo nutné ošetřit prodlení při kliknutí pomocí knihovny `fastclick` [11] a pro výběr barev kartiček je využita knihovna `spectrum` [9]

O obsluhu `WebView` se starají 3 hlavní soubory.

- a) ***connectionPlugin.js*** tento soubor má na starosti komunikaci s Javou. Po přijetí zprávy ji zpracuje a zavolá příslušné metody uložené v souboru `main.js`
- b) ***main.js*** je hlavní soubor, který má na starosti obsluhovat události všech tlačítek a přijímat zprávy od souboru `connectionPlugin.js`
- c) ***cards.js*** V tomto souboru jsou uloženy kombinace jednotlivých karet a další potřebné funkce pro obsluhu kartiček.

Dále má každá kartička svůj Javascriptový soubor, kde je uložen polygon každého obrázku generovaný pomocí online aplikace `Summer` [10]. Ukázka vytvoření mapy jedné konkrétní kartičky je na obrázku 5.3. Po kliknutí je tedy pomocí tohoto konkrétního souboru možno zjistit, na jaký obrázek uživatel skutečně klikl. Kartiček je ve hře celkem 55 a na každé je 8 obrázků. Z čehož každé dvě kartičky mají jeden stejný obrázek. Navíc jsou obrázky na každé kartičce různě velké.

## Knihovna jQuery

jQuery [7] je JavaScriptová knihovna, která nám usnadňuje psaní v JavaScriptu a je optimalizována pro velké množství prohlížečů. jQuery je otevřený a svobodný software pod licencí MIT, který se v posledních několika letech velice rozšířil. V celé aplikaci tedy pro práci s JavaScriptem používám tuto knihovnu.

## Knihovna jQuery mobile

jQuery mobile [2] je knihovna, která nám pomůže vytvořit responzivní design, který se jednoduše přizpůsobí na různé velikosti displeje. jQuery mobile využívá ke své práci knihovnu jQuery. Pyšní se podporou všech moderních desktopových prohlížečů, smartphonů, tabletů a e-čteček. jQuery Mobile nabízí pět barevných schémat. Pokud nám žádné nevyhovuje, můžeme si vzhled kompletně navrhnout znovu – pomocí ThemeRolleru [13], který vývojáři připravili speciálně pro jQuery Mobile. Práce s ním je velice jednoduchá a instinktivní, pro navržení šablony se standardně použije HTML5 a konkrétní prvky jQuery mobile se uvádějí do atributu `data`. Na obrázku 5.8 můžete vidět použití jQuery mobile v aplikaci Dobble. Knihovnu použijeme, tak že do atributu `data` vložíme potřebné informace, pokud například chceme nějakému elementu přiřadit roli, jednoduše do `data-role` vložíme hodnotu požadované role `navbar`. Další možností je zvolit si barevné schéma tlačítka tak, že do atributu `data-theme` vložíme jednu z předdefinovaných barevných schémat A až E.

```
<div data-role="navbar" id="mainpagenavbutton">
  <ul>
    <li>
      <button data-theme="a" id="singleplayer-mainpagenavbutton">
        Single player
        <br/>
        (time attack)
      </button>
    </li>
  </ul>
</div>
```

Obrázek 5.8: Použití jQuery mobile v aplikaci Dobble

## Knihovna FastClick

FastClick je easy-to-use knihovna pro odstranění 300ms prodlevy při klepnutí na displej. Jelikož je hra napsána ve frameworku PhoneGap, znamená to ve

skutečnosti, že hra je běží v mobilním prohlížeči. A to sebou přináší své výhody, například přenositelnost mezi zařízeními, ale také nevýhody například 300 milisekundovou prodlevu při doteku displeje. Tato prodleva má samozřejmě své opodstatnění. Pokud si v běžném prohlížeči zobrazíte stránku, která není optimalizována pro mobilní zařízení, je vidět velice nečitelně. Proto stačí dvakrát kliknout na displej a prohlížeč automaticky přiblíží zobrazený obsah. A právě daný dvojitý klik způsobuje tuto prodlevu. Aby prohlížeč rozpoznal, že jsme skutečně poklepali na displej zařízení dvakrát, musí po prvním kliknutí počkat 300 milisekund jestli nepříjde i druhý klik, který by znamenal přiblížení zobrazeného obsahu. Ale hra Dobble je optimalizována pro mobilní zařízení, takže dvojitý klik pro přiblížení nepotřebujeme. Přináší nám tedy jen zpoždění po kliknutí a vytváří iluzi zpomalenosti. Tato knihovna tedy dokonale řeší náš problém. Odstraňuje možnost dvojího kliknutí na displej a proto zařízení po prvním doteku displeje provede událost spojenou s kliknutím a nečeká 300 milisekund na druhé kliknutí.

### 5.1.7 HTML5 a CSS3

HTML5 a CSS3 [4] jsou více než jen dva nové standardy představené W3C. Slouží hlavně k tomu, aby nám pomohli tvořit nové lepší moderní webové aplikace. Aplikace tvořené HTML5 se snadněji vyvíjejí a spravují a také jsou přívětivější k uživatelům. HTML5 obsahuje nové elementy a CSS3 poskytuje nové pokročilé selektory, grafická vylepšení a lepší podporu fontů. Proto není již nutné jednodušší animace implementovat pomocí javascriptu.

#### a) Úložiště

Aby bylo možné vytvářet plnohodnotné aplikace, které fungují pouze na straně klienta, nesmí chybět úložiště, ve kterém si můžeme uchovat data. V HTML5 můžeme využívat webové SQL databáze, nebo localStorage. Lokální SQL databáze v Dobble využity nejsou, ale localStorage nám slouží například pro ukládání nastavení. LocalStorage ukládá data v párech klíč/hodnota.

#### b) Formuláře

Dříve jsme museli používat JavaScript a CSS k vytvoření posuvných jezdců, kalendáře pro výběr data a kapátka pro výběr barvy. Všechny tyto prvky



jsou nyní reálně definovány elementy HTML5, podobně jako výběrové nabídky, zatrhávací políčka nebo přepínače. Dalším přínosem jsou formuláře typu email, číslo, telefon nebo url. Přináší nám to například velkou výhodu právě u mobilních zařízení, protože prohlížeč nám může zobrazit pouze číselnou klávesnici pro zadání telefonního čísla a navíc se formuláře automaticky validují a nedovolí odeslání formuláře, pokud není zadán například validní email ve formuláři typu email.

### c) Pokročilé CSS selektory

CSS3 obsahuje nové selektory, které nám umožňují například identifikovat liché a sudé řádky tabulky, všechna vybraná zaškrťovací políčka, nebo dokonce poslední odstavec ve skupině. Proto není již nutné "obcházet" CSS3, abychom se dostali v efektům, které potřebujeme získat. Bylo potřeba použít JavaScript nebo bylo nutné přidávat třídy do HTML, které se poté stylovaly.

### d) Vizualní efekty a animace

Stíny za obrázky, tlačítka nebo texty pomáhají dostat do aplikací hloubku a přechody také mohou přidat na efektu prostoru. CSS3 nám umožňuje přidávat stíny a přechody elementům bez nutnosti spoléhat na obrázky. Navíc je možné používat transformace k vytváření kulatých rohů nebo ke zkosení či otočení elementů. Kulaté kartičky jsou právě vytvořeny pomocí těchto prvků. Jedná se o element `div`, který má nastavený `border-radius` na vysokou hodnotu, takže maximální zkosení rohů způsobí, že výsledný tvar je kruh. Vytvoření animací je nyní s CSS3 také velice jednoduché, například pokud klikneme v naší hře na špatný obrázek, tak vyvoláme efekt, který způsobí, že pozadí se rozblíká červeně. K tomuto efektu stačilo pár řádek v CSS3 viz obrázek 5.9. Na obrázku můžeme vidět animaci, která je vázána na třídu `highlight-error` první parametr je název animace, která se má vykonat a druhý parametr, jak dlouho animace bude trvat. Definice animace se uvádí v procentech. Znamená to v jakém úseku se mají dané příkazy vykonat. V naší ukázkové animaci to tedy znamená, že ihned po spuštění se nastaví barva pozadí na červenou a v 5% celkového času animace se nastaví na zcela průhlednou a tak dále. Takže v celkovém výsledku vyvolá výstražný blikající efekt, který je pro nás důležitý pokud uživatel klikne na špatný obrázek.

```
.highlight-error {  
  animation: animace 1s;  
}  
@-webkit-keyframes animace {  
  0% { background-color: red; }  
  5% { background-color: rgba(0,0,0,0); }  
  10% { background-color: red; }  
  15% { background-color: rgba(0,0,0,0); }  
  20% { background-color: red; }  
  100% { background-color: rgba(0,0,0,0); }  
}
```

Obrázek 5.9: Znázornění animace v CSS3

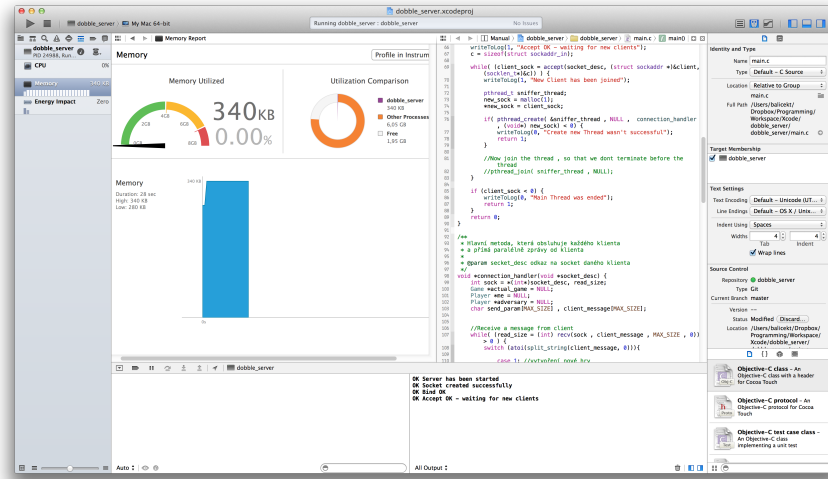
## 5.2 Server

Server nám slouží pro komunikaci mezi zařízeními. Je tedy potřeba, pokud se hráč rozhodne spustit multiuživatelskou hru online. Pro další dva herní módy potřeba není. Server byl napsán v rámci semestrální práce předmětu KIV/UPS - úvod do počítačových sítí a dále poupraven a doladěn v rámci bakalářské práce. Je naprogramován v jazyce C, kódován a odladěn ve vývojovém prostředí Xcode viz obrázek 5.10 od firmy Apple. Po připojení nového klienta se vytvoří nové vlákno, které se po celou dobu spojení o tohoto klienta stará.

Pro ukládání jednotlivých her je používá nový datový typ Game a jednotlivé hry jsou uloženy v spojovém seznamu. Pokud se odpojuje poslední hráč ze hry, daná hra se odstraní ze spojového seznamu a již není možné do ní vstoupit.

Server má tři hlavní soubory *.c* a ke každému hlavičkový soubor *.h* s hlavičkami metod. V souboru *main.h* jsou includovány potřebné knihovny a vytvořeny nové datové typy.

- Hlavní soubor *main.c* kde se nachází metoda main. A metoda pro vytvoření nového vlákna a obsluhu klienta.
- Soubor *game.c* kde jsou umístěny metody pro práci se hrou. Vytvoření, odstranění nové hry. Přidání a odstranění hráče ze hry.
- V souboru *function.c* se nachází pomocné metody. Například metoda pro parsování řetězce, metoda pro tisk nápovědy, metoda pro zpracování parametrů.



Obrázek 5.10: Vývojové prostředí Xcode

## 5.3 Komunikační protokol

Pro komunikaci mezi klientem a serverem je použit TCP protokol (Transmission Control Protocol). Díky tomuto protokolu může server a klient mezi sebou vytvořit spojení přes které poté přenáší data. TCP zaručuje spolehlivé doručení a doručení ve správném pořadí.

### 5.3.1 Zahájení komunikace serveru

- socket - Vytvoří socket daného typu. Socket není asociován s žádným portem.
- bind - Naváže socket se jménem. Používá pouze server.
- listen - Slouží k nastavení socketu do stavu čekání na příchozí spojení.
- accept - Funkce vrátí parametry prvního spojení ve frontě a nový socket, který slouží ke komunikaci s druhou stranou. Původní spojení ve frontě a nový socket, který slouží ke komunikaci.
- recv - Příjem zpráv ze socketu.
- send - Odeslání zprávy.

### 5.3.2 Zahájení komunikace klienta

- a) socket - Vytvoří socket daného typu. Socket není asociován s žádným portem.
- b) connect - Pro navázání spojení. Používá pouze klient.
- c) send - Odeslání zprávy. Lze použít pouze pokud je socket ve stavu připojen.
- d) recv - Příjem zpráv ze socketu.
- e) close - Uzavření socketu.

### 5.3.3 Komunikace mezi serverem a klientem

Klient a server se dorozumívají pomocí číselných signálů viz následující tabulka. Pokud je potřeba přijmout nebo odeslat nějaký parametr, tak za číselným kódem odděleno středníkem je umístěn další parametr. Například pokud server posílá informaci o nově vygenerovaných kartách, kód který přijde klientovi vypadá následovně 3;16;43 první číslo určuje, o který signál se jedná a druhá dvě čísla jsou již parametry tedy číslo karty 1 a číslo karty 2.

Přehled zasílaných kódů:

Klient -> Server

- 1 - vytvoření nové hry;počet vítězství na hru
- 2 - Připojení se do již vytvořené hry podle ;ID hry
- 3 - Zaslání informace o správné odpovědi
- 4 - Zaslání informace o špatné odpovědi
- 5 - Hráč je připraven ke hře
- 6 - Hráč není připraven ke hře
- 7 - Odpojit hráče ze hry

Server -> Klient

- 1 - Vytvoření nové hry proběhlo v pořádku ;ID hry;karta 1;karta 2

- 2 - Vytvoření nové hry se nezdařilo
- 3 - Připojení do nové hry proběhlo v pořádku ;karta 1;karta 2
- 4 - Připojení do nové hry se nezdařilo - tato hra neexistuje
- 5 - Připojení do nové hry se nezdařilo - žádné hry nejsou vytvořeny
- 6 - Chyba volaná služba neexistuje
- 7 - Připojení do nové hry se nezdařilo - hra je obsazena
- 8 - Do hry se připojil protihráč
- 9 - Připojení do nové hry se nezdařilo - počet hráčů ve hře je vyčerpán (hra nemá volný socket)
- 10 - Protihráč se odpojil z této hry
- 12 - Spojení bylo navázáno
- 14 - Spojení se nepodařilo vytvořit
- 15 - Spojení se serverem selhalo
- 16 - Vytvoření nebo připojení k nové hře se nezdařilo Klient je již v nějaké hře připojen;ID hry
- 17 - Nové skóre já ;skóre
- 18 - Nové skóre protivník ;skóre
- 19 - Nová čísla karet, nové skóre já ;skóre;karta 1;karta 2
- 20 - Nová čísla karet, nové skóre protivník ;skóre;karta 1;karta 2
- 21 - Oba hráči ready - start hry
- 22 - Konec hry já jsem vyhrál
- 23 - Konec hry protivní vyhrál

## 5.4 Časově kritické úseky hry

Časově kritické úseky hry mohou nastat pouze v herním režimu multiplayer online. Ve verzi offline, kdy uživatelé hrají na jednom zařízení budou mít vždy stejné podmínky. Pokud bereme v úvahu rychlost odezvy systému nebo dobu potřebnou na vyhodnocení správnosti kliknutí. Když se podíváme na hru online, kde každý uživatel má své vlastní zařízení, je zde již několik aspektů, které by nám mohli hru ovlivňovat.

### 5.4.1 S ohledem na výkonnost zařízení

První by mohla být výkonnost daného zařízení, která by mohla prodloužit dobu potřebnou pro zjištění, na který obrázek bylo kliknuto a porovnání jestli je zvolené rozhodnutí správně. Ale vzhledem k tomu, že tyto úkony jsou opravdu velice rychlé, tak výkonnost zařízení je zanedbatelná a výsledek hry nám prakticky neovlivní, pokud to nebude extrémně pomalé zařízení. Hra byla testována i na emulátoru, který bych nazval opravdu extrémně pomalý a pomalejší reálné zařízení budeme hledat složitě. Přesto byla hra hratelná a rozdíl v porovnání s výkonnějším reálným zařízením je určitě znát, ale stále to je v přijatelné míře.

### 5.4.2 S ohledem na rychlost připojení

Druhým a výrazně závažnějším problémem může být rychlost připojení k síti. Data, která se přenáší mezi serverem a zařízením jsou minimalizována a jak bylo zmíněno výše jedná se pouze o číslo zprávy, které se má vykonat. Kdybychom vzali v úvahu, že jeden z uživatelů bude v nejrychlejší síti LTE, kde se rychlost stahování pohybuje průměrně v řádech desítek Mb/s a druhý uživatel v odlehlejší oblasti s pokrytím EDGE s průměrnou rychlostí stahování cca 150 kb/s. Tento problém nám již může přinést značné komplikace. Pokud vezmeme v úvahu, že oba hráči se dotknou displeje zařízení současně, tak je dá očekávat, že hráč s rychlejším připojením k síti získá určitou výhodu. V této situaci by se dal tento hendikep ještě zanedbat, protože k naprosto totožnému stisknutí displeje dojít prakticky nemůže. Může nastat, ale závažnější situace. Hráč s pomalejším připojením vidí správnou odpověď dříve, ale než doputuje jeho správná odpověď k serveru předstihne ho druhý hráč s rychlejším připojením. Toto už je mnohem závažnější problém a v následující kapitole se podíváme na jeho možná řešení.

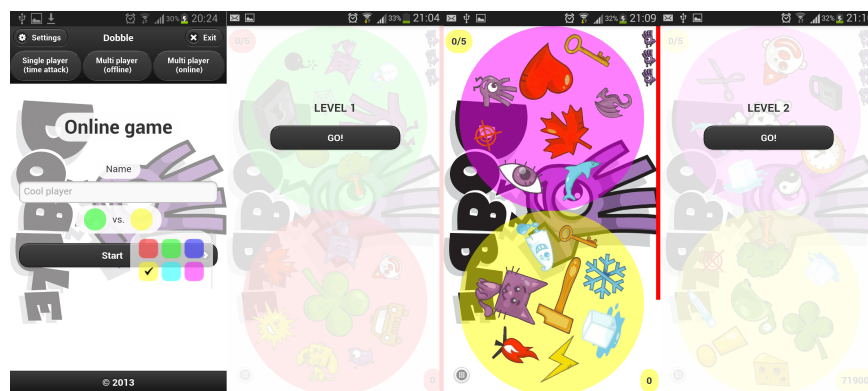
### 5.4.3 Možná řešení

Jedním z řešení by mohlo být zasílání časových značek spolu s odesláním odpovědi na server. Zde ale narazíme na synchronizaci času obou zařízení. Synchronizace by se dala vyřešit tak, že při připojení klienta na server by mu server zpět zaslal svůj aktuální čas, od kterého by se poté nová hra odvíjela. Pokud bychom přešli na takovéto řešení, server by po přijetí zprávy

uložil čas správné odpovědi a pokud by přišla zpráva od druhého uživatele se stejnými daty ke stejným kartám se správnou odpovědí, ale s nižší časovou značkou body by se přičetli hráči, od kterého přišla odpověď na server déle, ale vzhledem k časové značce byla odeslána z daného zařízení dříve. To znamená, že uživatel byl rychlejší, ale prodlení nebylo způsobeno jeho vinnou, ale rychlostí připojení a to je pro nás důležitý aspekt.

## 6 Uživatelská dokumentace

Hra Dobble se ovládá pouze dotyky displeje. Po spuštění hry je defaultně nastaven layout portrét, tedy zobrazení na výšku. Na událost překlopení na šířku tedy zařízení nijak nereaguje.



Obrázek 6.1: Single player

### 6.1 Single player

Daný popis se vztahuje k obrázku 6.1

- Před spuštěním hry je možné si nastavit barvu kartiček a jméno hráče.
- Hráč vstoupí do 1 levelu, kde je nastaven maximální časový limit a v každém dalším kole se zrychluje.
- Po 5 správných uhodnutí stejných obrázků na zobrazených kartičkách hráč postoupí do dalšího kola

### 6.2 Multi player - offline

- Před zahájením hry se nastaví počet bodů, které je potřeba získat k vítězství ve hře, jména jednotlivých hráčů a barvy kartiček



b) Hráč, který nejdříve nasbírá stanovený počet bodů, vyhrává.



Obrázek 6.2: Postup vytvoření nové online hry

## 6.3 Multi player - online

Spuštění hry probíhá následovně viz obrázek 6.2.

a) Úvodní stránka po spuštění aplikace Dobble

- b) Nastavení IP adresy a portu serveru
- c) Záložka nastavení online hry - Zde je možnost nastavit si barvu kartiček
- d) Vytvoření nebo připojení do nové hry, možnost nastavení maximálního počtu vítězství na jednu hru - V horní části okna musí být červený nápis *Connection OK* pokud je tam například *Connection Error* nepodařilo se připojit k serveru a tím pádem nepůjde vytvořit nová hra ani se k jiné hře připojit.
- e) Čekání na připojení protihráče - Po připojení protihráče se tlačítko start přepne do samo do stavu *actived* a je možné na něj kliknout a zahájit novou hru.
- f) Samotná hra Dobble - Klikání na obrázky je umožněno pouze na spodní kartičce na obrázky číslo 4 tedy červené.

## 6.4 Přeložení aplikace a spuštění aplikace

Přeložení serveru je velice snadné pomocí *make*. Je k dispozici i *make clean* pro smazání již vytvořeného spustitelného souboru. Poté se vygenerovaný skript spustí *./název\_skriptu*. K dispozici jsou následující parametry. *-help* pro vypsaní nápovědy, *-port* pro nastavení čísla portu a *-ip* pro nastavení ip adresy.

Pro spuštění klienta se musí nainstalovat příložený balíček s příponou *.apk*, tedy standardní instalační soubor pro zařízení s OS Android.

## 7 Distribuce hry

Abychom mohli nějakou mobilní aplikaci pro Android distribuovat, potřebujeme ji nejdříve exportovat do apk souboru. Pokud tedy chceme náš aplikační soubor vyexportovat pro použití na jiných zařízeních, pomůže nám k tomu Eclipse. Z nabídky zvolíme položku Android Tools -> Export Unsigned Application Package, která nám umožní exportovat soubor do námi zvoleného umístění. Tento soubor nebude digitálně podepsaný, proto nebude možná instalace na zařízení, které nebudou mít v telefonu povolenou volbu povolit instalaci z neověřených zdrojů.

### Vytvoření certifikátu

Pokud bychom chtěli aplikaci umístit do android marketu, je nutné ji mít digitálně podepsanou tzn. musíme vytvořit tzv. keysotre certifikát. Kterým naši aplikaci podepíšeme a bude nám umožněno nahrát ji do Android marketu. Postup pro vytvoření certifikátu je následující.

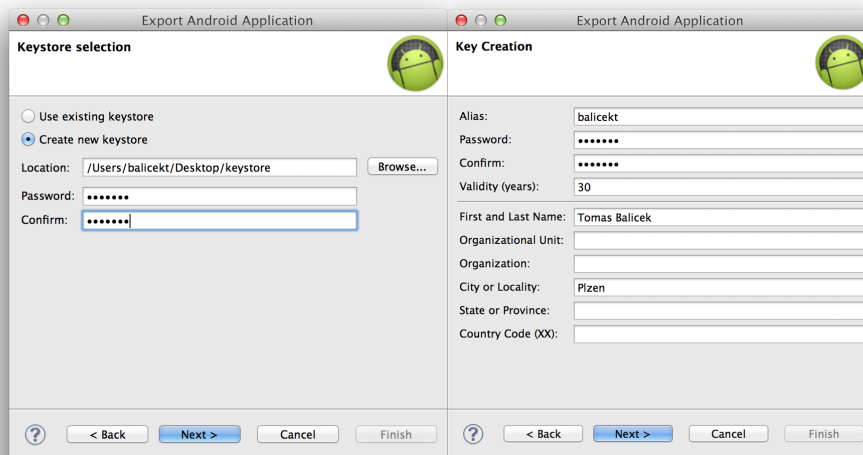
- 1) Pravým tlačítkem myši klikneme na ikonku našeho projektu.
- 2) Z nabídky zvolíme Android Tools -> Export Signed Application Package.
- 3) Poté už jen zvolíme náš projekt a vyplníme umístění, heslo a údaje o sobě jak ukazuje obrázek 7.1

Tímto jsme vygenerovali digitálně podepsaný apk soubor, který již můžeme nahrát do Android marketu.

### Upload aplikace do Android marketu

Pokud chceme hru publikovat na Android marketu, musíme být registrovaný Android vývojář a to vyžaduje zaplatit poplatek ve výši 25\$ a mít vlastní Google účet. Celý proces je relativně jednoduchý, musíme souhlasit s licenčními podmínkami a zaplatit výše zmíněný poplatek skrze Google checkout. Samotný upload je velice jednoduchý. Aplikaci ověřuje interní automatický systém a aplikaci je možno do Android marketu dostat během několika minut. Postup nahrání aplikace je následující:

- 1) Přejdeme na adresu *http://market.android.com*
- 2) Klikneme na tlačítko Upload Application, kde do příslušného formuláře vyplníme informace o aplikaci a přiložíme samotnou aplikaci. Maximální velikost nahrávané aplikace je 50MB.
- 3) Android market vyžaduje minimálně dva screenshoty naší aplikace, pak ikonku aplikace a další náležitosti, které se budou zobrazovat v rámci Android marketu
- 4) Před nahráním aplikace je potřeba nastavit atribut android:debuggable na false. Pokud bychom na to zapomněli, při nahrávání budeme upozorněni.
- 5) Po úspěšném nahrání se aplikace objeví v Android marketu během chvíle.



Obrázek 7.1: Vyplnění údajů pro nový keystore

## 8 Testování

Pokud vytváříme nějakou složitější aplikaci, dříve nebo později narazíme na otázku jak efektivně aplikaci testovat. Při vývoji pro PhoneGap můžeme pracovat s velkou řadou testovacích nástrojů já jsem využíval následující.

### 8.1 Ladící konzole Eclipse

V Eclipse jsem využíval možnost logování do ladící konzole pomocí příkazů viz obrázek 8.1. Toto není úplně nejlepší způsob testování, ale na základní věci to dostačuje. Výhodou těchto výpisů je, že se zobrazují pouze do konzole Eclipse a nemusí se při vystavení ostré verze z kódu nijak odstraňovat. Tyto příkazy si můžeme filtrovat v rámci nástroje LogCat, na který se můžeme přepnout zvolením jiné perspektivy v Eclipse a zvolením položky DDMS.

```
console.log("Logovací zpráva");  
console.warn("Varovací zpráva");  
console.error("Chybová zpráva");
```

Obrázek 8.1: Ladící výpisy konzole Eclipse

### 8.2 Ladící nástroje v prohlížeči

Můžeme využít hlavní výhodu PhoneGap a to, že se jedná o webovou aplikaci a testovat přímo v prohlížečích viz obrázek 8.3. Pro testování jsem používal prohlížeč Google Chrome. Chrome nabízí mnoho vývojářských nástrojů a pro programování, respektive testování webových aplikací, je podle mého názoru zdaleka nejlepší. Velkou výhodou vidím hlavně v tom, že webový prohlížeč se nachází v každém PC a bez žádných složitých instalací, nám poskytuje hodně možností. Můžeme se snadno dívat na obsah jednotlivých elementů, přidávat zarážky pro debugování kódu. Jedna z hlavních výhod, kterou jsem pro testování využíval je emulace mobilního zařízení, kde se po aktivaci prohlížeč chová jako mobilní zařízení. Můžeme si zde vybrat z velkého množství zařízení jako například Apple iPad, Apple iPhone, Amazon Kindle, Sony Xperia,

HTC, LG a velké množství telefonů od Korejského výrobce Samsung Galaxy. Na obrázku 8.3 můžeme vidět emulaci mobilního zařízení Apple iPhone 4. Pokud si zvolíme přímo konkrétní zařízení, Chrome za nás nastaví viewport i User Agenta. Pokud ale nechceme konkrétní zařízení, můžeme si všechny parametry nastavit sami. V záložce screen si můžeme nastavit rozlišení, které požadujeme a device pixel ratio takzvaný pixelový poměr. V záložce user agent si můžeme nastavit konkrétní řetězec a v záložce sensors geolokaci a akcelerometr.

### Emulace viewportu a device pixel ratio

V záložce screen si můžeme nastavit hodnotu rozlišení podle naší libosti. A tak si snadno testovat, jak se naše aplikace chová v zařízeních s různým rozlišením. Device pixel ratio je takzvaný poměr mezi fyzickým a abstraktním rozlišením zařízení. Pokusím se to vysvětlit na příkladu. Vezmeme si Samsung Galaxy S4, který má rozlišení Full HD 1920x1080px, tudíž má vysoké DPI neboli počet zobrazených bodů na jednotku jednoho palce. Musí se toto vysoké rozlišení nějakým způsobem redukovat a k tomu nám slouží výše zmíněné device pixel ratio. Je to číselný údaj, který udává kolikrát má být fyzické rozlišení zmenšeno. Pokud tedy vezmeme v úvahu S4 od Samsungu má rozlišení na šířku 1080px a device pixel ratio má hodnotu 3. To znamená, že zařízení bude zobrazovat obsah jako by měl rozlišení 360px. Nastává tedy otázka, proč výrobci dělají takto vysoké rozlišení, které nakonec není ani využito. Vysvětlení je jednoduché, i když vysoké rozlišení není fyzicky využito, tak je to na displeji opravdu znát, pokud se na něj podíváme zblízka, na první pohled poznáme, že displej má vysoké rozlišení. Zobrazení na displeji je opravdu jemné a kvalitní. Důkazem jsou vysoce kvalitní displeje retina od firmy Apple. Například od 13 palcového macBooku Pro umístí displej retina s rozlišením 2560x1600. Takové rozlišení nemá v dnešní době ani většina vyráběných televizorů s úhlopříčkou přes 100cm. Ve PhoneGapu nebo webových aplikacích můžeme práci s rozlišením velice dobře využít, pokud například děláme responzivní webdesign. Jednoduše řečeno chceme aplikaci, která se bude pěkně zobrazovat jak na 25 palcovém monitoru, tak na telefonu s velikostí displeje 4 palce. Pro práci použijeme takzvané CSS media queries. Jsou to dotazy na určité parametry. Pomocí těchto dotazů si snadno můžeme zjistit například to, jestli je rozlišení menší než 400px a upravit podle toho vzhled a přizpůsobit ho mobilním zařízením. Pokud se dotazujeme právě na rozlišení, je již zpracováno pomocí výše zmíněného device pixel ratio. Takže velice snadno zjistíme, jestli se jedná o mobilní zařízení.

## Emulace user agenta

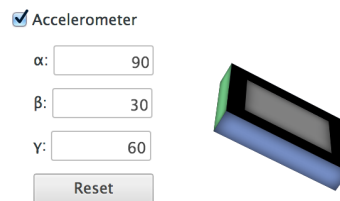
User agent je řetězec, který se odesílá v hlavičce požadavku o nějaká data na server. Z tohoto řetězce rozpoznáme několik důležitých informací o zařízení, které o data žádá. Řetězec může vypadat například takto:

```
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.36  
(KHTML, like Gecko) Chrome/34.0.1847.116 Safari/537.36
```

Z tohoto řetězce poznáme, že se jedná o PC s operačním systémem Mac OS X 10.9.2 a používá prohlížeč Google Chrome 34.0.1847.116, který má jádro AppleWebKit.

## Emulace geolokace a akcelerometru

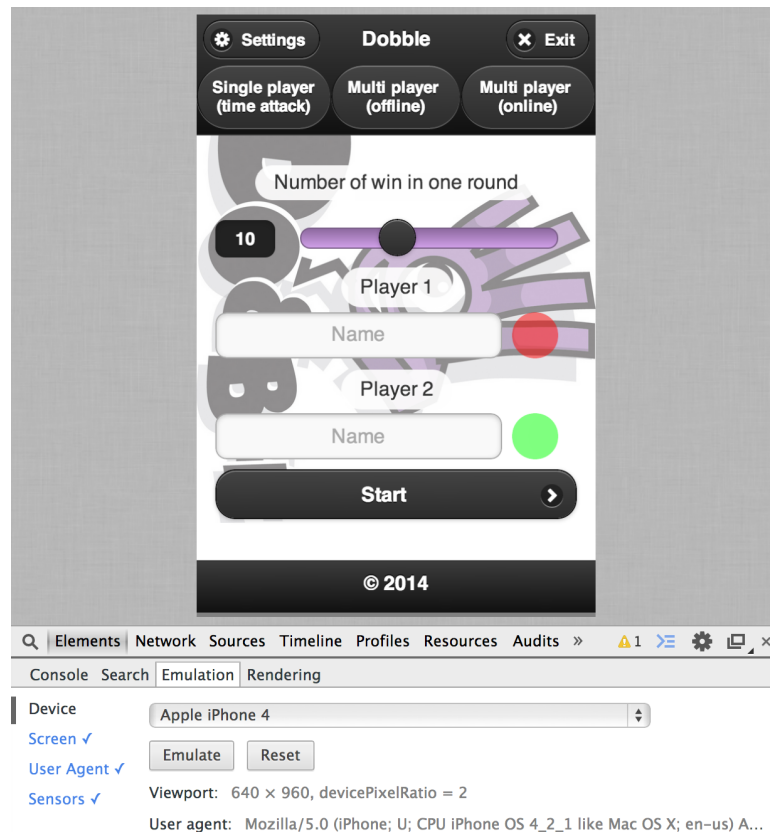
V poslední záložce máme možnost si nastavit geolokaci a souřadnice akcelerometru. U geolokace si můžeme nastavit přesné souřadnice, kde se nacházíme nebo zaškrtnout, že je pozice nedostupná, abychom si otestovali, jak se dané zařízení bude chovat, pokud například v průběhu ztratí signál. Další velice zajímavou možností pro testování, je nastavení souřadnic akcelerometru, které můžeme vidět na obrázku 8.2. Zde si můžeme dobře nasimulovat naklonění zařízení a snadno tak otestovat chování aplikace. Pokud v naší aplikaci tento senzor využíváme.



Obrázek 8.2: Ukázka nastavení souřadnic akcelerometru

## 8.3 Emulátor

Testování v emulátoru je poněkud zdlouhavé z důvodu vysoké náročnosti na hardware. Emulátor byl spouštěn na platformě s unixovým jádrem Mac



Obrázek 8.3: Emulace zařízení Apple iPhone 4 v prohlížeči Google Chrome

OS X obrázek 8.4, i na Windowsech. Pokud testujeme v emulátoru, můžeme využívat ladící konzoli Eclipse a testovat již plnou funkčnost hry.

## 8.4 Zařízení

Testování v reálném zařízení je o hodně svižnější než v emulátoru. Pouze první spuštění je komplikovanější. Před nahráním na reálný telefon se musí provést několik kroků. Pokud chceme hru testovat na zařízení Android, nemusíme mít placený vývojářský certifikát, jako je tomu například u iPhone. U OS Android tedy provedeme následující nastavení: V nastavení telefonu přejdeme do karty Aplikace -> Vývoj a zatrhneme volbu Ladění USB (text se může drobně lišit v závislosti na verzi).





Obrázek 8.4: Hra Dobble spuštěná v emulátoru na platformě OS X 10.9 Mavericks

Poté stačí již jen propojit telefon USB kabelem s PC a doinstalovat potřebné ovladače. Při spuštění aplikace v Eclipse by nám měl dát na výběr jestli chceme aplikaci spustit v emulátoru nebo na reálném zařízení.

### Testovaná zařízení

Testování aplikace probíhalo na následujících reálných zařízeních:

Samsung Galaxy SI i9000 - verze API 2.3.6 rozlišení 800x480

Samsung Galaxy SIII i9300 - verze API 4.0, rozlišení 1280x720px.

Samsung Galaxy S4 i9505 - verze API 4.2, rozlišení 1920x1080px  
HTC One V - Verze API 4.0.3, rozlišení 800x480

Na všech výše uvedených zařízeních hra fungovala bez problémů. Hra byla již předvedena v rámci Dne otevřených dveří na fakultě FAV katedry informatiky a návštěvníci ji měli zpřístupněnou, jak ve formě emulátoru, tak na reálném zařízení.

## **8.5 Online verze**

Testování online verze požaduje navíc spuštěný server. Aby testování bylo opravdu důkladné, bylo by potřeba větší množství zařízení. Testy serveru probíhali na školních strojích, konkrétně students.kiv.zcu.cz, na které je možno přistupovat z vnější sítě. Po spuštění serveru jsem postupně připojoval zařízení a testoval různé kombinace a propojení jednotlivých zařízení. Testy dopadly výborně, hra byla plně hratelná.

## **8.6 Test s ohledem na rychlost připojení**

Rychlost připojení byla testována na dvou zařízeních, kde jedno bylo připojeno do sítě pomocí Wi-Fi a druhé pomocí pomalejšího mobilního připojení. I přes různé rychlosti připojení se nepovedlo nasimulovat, aby hráč s pomalejším zařízením byl znevýhodněn. Prodlevy odpovědí ze serveru byly opravdu zanedbatelné, takže znevýhodnění pomalejšího připojení se nepovedlo prokázat.

## 9 Závěr

Cílem této bakalářské práce bylo navržení, realizace a nakonec samotné ověření funkcionality hry Dobble.

Do současné doby nebyla tato hra dostupná pro prostředí Android, ale pouze pro iOS a to bez herního módu multiplayer online.

V prvních kapitolách jsem se věnoval prozkoumávání jednotlivých mobilních zařízení a jejich možnému použití pro mobilní hry, potažmo multiuživatelské hry online. Dále byla hra navržena po vzoru papírové verze hry Dobble a realizována pomocí multiplatformního frameworku PhoneGap. Nakonec, aby byla ověřena funkcionality, bylo potřeba hru otestovat na různých zařízeních s rozdílnou verzí API a rozlišením displeje. Funkcionality hry byla ověřena ve všech 3 uvedených herních scénářích a to jak v emulátoru, tak v reálných zařízeních. Jak už bylo zmíněno velkou výhodou hry je, že je napsána ve frameworku PhoneGap, proto ji můžeme snadno přenést na další platformy.

V praxi se ukázalo, že pomocí multiplatformních nástrojů se dá vyvinout plnohodnotná aplikace, potažmo hra. Webové technologie čeká díky HTML5 a CSS3 v následujících letech další pokrok, proto i v oblasti multiplatformních her můžeme očekávat novinky.

V blízké době plánuji další úpravy, v první řadě umožnit hrát hru i uživatelům iPhone a také přímo ve webovém prohlížeči. Za další dodělat statistiky pro single player, kde by bylo vidět nejvýše nahrané skóre a další zajímavé informace. Dále možnost zadávání jména každého hráče a sledování statistiky mezi jednotlivými hráči. A nakonec přihlašování hráčů pomocí Facebooku, Twitteru popřípadě Google+.

# Seznam obrázků

2.1	Znázornění gyroskopu [14]	4
2.2	Princip rezistivního displej [12]	5
3.1	Komunikace zařízení přes server	8
5.1	Hra Dobble v1.0	11
5.2	Ukázka shodných obrázků na kartičkách	13
5.3	Generování polygonů jedné kartičky	14
5.4	Vygenerovaná HTML mapa	14
5.5	Tabulka podporovaných vlastností jednotlivých platforem	16
5.6	Struktura Android aplikace s přidáním frameworkem PhoneGap	17
5.7	Metoda onCreate(), která je srdcem celé aplikace	18
5.8	Použití jQuery mobile v aplikaci Dobble	20
5.9	Znázornění animace v CSS3	23
5.10	Vývojové prostředí Xcode	24
6.1	Single player	29
6.2	Postup vytvoření nové online hry	30

7.1	Vyplnění údajů pro nový keystore . . . . .	33
8.1	Ladící výpisy konzole Eclipse . . . . .	34
8.2	Ukázka nastavení souřadnic akcelerometru . . . . .	36
8.3	Emulace zařízení Apple iPhone 4 v prohlížeči Google Chrome	37
8.4	Hra Dobble spuštěná v emulátoru na platformě OS X 10.9 Mavericks . . . . .	38

# Přehled použitých zkratk a značení

API	Application Programming Interface. Rozhraní pro vývoj aplikací.
Bluetooth	Standard pro bezdrátovou komunikaci.
HTML	HyperText Markup Language. Jazyk pro tvorbu webových stránek.
CSS (CSS3)	Cascading Style Sheets. Kaskádové styly jsou jazyk pro popis zobrazení webových stránek.
IEEE	Institute of Electrical and Electronics Engineers. Je mezinárodní nezisková profesní organizace usilující o vzestup technologie související s elektrotechnikou.
Wi-Fi	Bezdrátová komunikace v počítačových sítích.
W3C	World Wide Web Consortium, jehož členové vyvíjejí webové standardy pro World Wide Web.
PAN	Personal Area Network. Jsou sítě s nejmenší rozlehlostí, používané pro propojení osobních elektronických zařízení typu mobilní telefon, laptop nebo PDA.
Cardanův závěs	závěs se dvěma kruhy volně pohyblivými kolem os k sobě kolmých.
GUI	Graphical User Interface. Uživatelské rozhraní, které umožňuje ovládat počítač pomocí interaktivních grafických ovládacích prvků.

JS	JavaScript. Multiplatformní, objektově orientovaný skriptovací jazyk.
WP Store	Windows Phone Store. Obchod s aplikacemi pro uživatele zařízení na platformě Windows Phone.
OS	Operační systém. Základní programové vybavení počítače.
SDK	Software development kit. Sada softwarových vývojových nástrojů, která nám umožňuje vytvářet jisté aplikace.
ADT	Android Development Tools. Jedná se o plugin pro Eclipse pomocí, kterého získáme prostředí pro vývoj Android aplikací.
IDE	Integrated Development Environment. Prostředí pro vývoj aplikací většinou zaměřené na konkrétní jazyk.
TCP	Transmission Control Protocol. Základní spolehlivý přenosový protokol transportní vrstvy.
IP	Internet Protocol. Základní protokol pracující na síťové vrstvě.
DPI	Dots per inch. Údaj určující, kolik obrazových bodů se vejde do délky jednoho palce.
USB	Universal Serial Bus. Moderní způsob připojení periferií k počítači.
QWERTY	Normalizované rozložení kláves na klávesnici.
LTE	3GPP Long Term Evolution - technologie pro vysokorychlostní internet v mobilních sítích.
EDGE	Enhanced Data rates for GSM Evolution - technologie pro přenos dat v sítích GSM.

GSM	Globální Systém pro Mobilní komunikaci.
iOS	je mobilní operační systém vytvořený společností Apple Inc.



# Literatura

- [1] JMiroslav Ujbányai *Programujeme pro Android* Grada Publishing, a.s., 2012. ISBN 978-80-247-3995-3.
- [2] Jiří Vávru *jQuery Mobile* Computer Press, a.s., 2013. ISBN 978-80-251-3811-3.
- [3] Den Odell *JavaScript* Computer Press, a.s., 2010. ISBN 978-80-251-2733-9.
- [4] Brian P. Hogan *HTML 5 a CSS 3* Computer Press, a.s., 2011. ISBN 978-80-251-3576-1.
- [5] Ondřej Baše *jQuery pro neprogramátory Průvodce využitím knihovny jQuery UI* Computer Press, a.s., 2012. ISBN 978-80-251-3750-5.
- [6] Adobe Systems Inc. *Framework pro mobilní zařízení* [online]. 2014. [cit. 6.1.2014]. Dostupné z: <http://phonegap.com>.
- [7] The jQuery Foundation. *FKnihovna pro usnadnění práce s javascriptem* [online]. 2014. [cit. 22.3.2014]. Dostupné z: <http://jquery.com>.
- [8] The jQuery Foundation. *User interface pro responzivní webové stránky* [online]. 2014. [cit. 22.3.2014]. Dostupné z: <http://jquerymobile.com>.
- [9] Brian Grinstead. *Knihovna pro možnost výběru barvy* [online]. [cit. 5.4.2014]. Dostupné z: <http://bgrins.github.io/spectrum>.
- [10] Vera Lobacheva. *Webová aplikace pro vytvoření souřadnicové mapy* [online]. [cit. 5.4.2014]. Dostupné z: <https://github.com/summerstyle/summer>.
- [11] FT Labs. *Knihovna pro snížení odezvy při kliknutí na displej* [online]. [cit. 29.3.2014]. Dostupné z: <https://github.com/ftlabs/fastclick>.

- [12] mobilmania.cz. *Už vím, jak fungují dotykové displeje* 2004 [online]. [cit. 27.2.2014]. Dostupné z: <http://www.mobilmania.cz/uz-vim-jak-funguji-dotykove-displeje/a-1108570/default.aspx>.
- [13] jQuery mobile. *Přizpůsobení barevného provedení jQuery mobile* [online]. [cit. 4.5.2014]. Dostupné z: <http://themroller.jquerymobile.com/>.
- [14] mobilenet.cz. *Senzory mobilních zařízení* 2013 [online]. [cit. 16.3.2014]. Dostupné z: <http://mobilenet.cz/clanky/techbox-vas-telefon-je-prospikovany-senzory-12496>.
- [15] Bluetooth.org. *Bluetooth specifikace* [online]. [cit. 7.5.2014]. Dostupné z: <https://www.bluetooth.org/en-us/specification/adopted-specifications>.