

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## PROHLÍŽEČ FOTOGRAFIÍ S LEAP MOTION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAL SANITRÁK

BRNO 2015



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## **PROHLÍŽEČ FOTOGRAFIÍ S LEAP MOTION**

PHOTO EXPLORER WITH LEAP MOTION

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MICHAL SANITRÁK**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. JIŘÍ ZAHRÁDKA**

BRNO 2015

## **Abstrakt**

Cílem této bakalářské práce bylo navrhnout a vytvořit aplikaci prohlížeče fotografií, který je možno ovládat pomocí gest rukou s využitím senzoru Leap Motion. Aplikace byla implementována v programovacím jazyce Java s využitím platformy JavaFX. Prohlížeč fotografií disponuje přehledným uživatelským rozhraním a je možné jej ovládat gesty rukou. Aplikace je multiplatformní a lze ji spustit na platformě Linux a Windows.

## **Abstract**

The goal of this thesis was to design and create photo explorer application, which can be controlled by hand gestures using the Leap Motion sensor. Application was implemented in Java programming language using the JavaFX platform. Photo explorer has a clear user interface and can be controlled with hand gestures. The application is multiplatform and can be run on Linux and Windows.

## **Klíčová slova**

Ovládání pohybem ruky, gesta rukou, Leap Motion, prohlížeč fotografií

## **Keywords**

Hand motion control, hand gestures, Leap Motion, photo explorer

## **Citace**

Michal Sanitrák: Prohlížeč fotografií s Leap Motion, bakalářská práce, Brno, FIT VUT v Brně, 2015

# Prohlížeč fotografií s Leap Motion

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jiřího Zahradky.

.....  
Michal Sanitrák  
19. května 2015

## Poděkování

Chtěl bych poděkovat panu Ing. Jiřímu Zahradkovi za vedení práce, ochotu a věcné připomínky. Dále bych chtěl poděkovat své rodině za podporu při vypracovávání této práce a také všem, kteří se zapojili do testování aplikace a poskytli mi tak cenné informace.

© Michal Sanitrák, 2015.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Detekce pohybu rukou a senzory pro detekci gest</b>	<b>3</b>
2.1	Současnost . . . . .	3
2.2	Popis gesta . . . . .	4
2.3	Metody detekce pohybu a gest . . . . .	4
2.4	Senzory pro detekci pohybu a gest . . . . .	6
<b>3</b>	<b>Prohlížeče fotografií</b>	<b>11</b>
3.1	Rozdělení a popis prohlížečů obrázků . . . . .	11
3.2	Prohlížeče obrázků ovládané gesty . . . . .	12
<b>4</b>	<b>Leap Motion SDK</b>	<b>14</b>
4.1	Leap Motion SDK . . . . .	14
<b>5</b>	<b>Návrh</b>	<b>17</b>
5.1	Popis návrhu . . . . .	17
5.2	Návrh funkcí a gest . . . . .	17
5.3	Grafické rozhraní . . . . .	20
5.4	Nastavení . . . . .	21
<b>6</b>	<b>Implementace</b>	<b>22</b>
6.1	Popis celkového vývoje . . . . .	22
6.2	Struktura projektu . . . . .	23
6.3	Gesta . . . . .	24
6.4	Nastavení . . . . .	24
6.5	Ukládání změn . . . . .	25
6.6	Grafické rozhraní . . . . .	25
<b>7</b>	<b>Testování</b>	<b>27</b>
7.1	Hodnocení dotazníkem . . . . .	27
7.2	Přímé testování . . . . .	29
<b>8</b>	<b>Závěr</b>	<b>31</b>
<b>A</b>	<b>Obsah CD</b>	<b>34</b>

# Kapitola 1

## Úvod

Již od počátků sci-fi filmů bylo velmi oblíbené využívání bezdotykového ovládání různých systémů. Jedním z nejnámějších filmů, který využívá tento druh ovládání je *Minority Report*. Sci-fi se však díky dnešním technologiím stává realitou, alespoň co se bezdotykového ovládání týče. Je tak možné jednoduše ovládat aplikace a různá zařízení pouze s využitím gest. To v sobě skýtá obrovské možnosti uplatnění. Začínají tak vznikat nová zařízení pro detekci gest a jedním z nich je právě senzor *Leap Motion*. Takováto zařízení jsou pak využívána nejen v herním průmyslu, kde si získávají velkou oblíbenost, ale také ve vědecké sféře.

Proto jsem si zvolil jako téma prohlížeč fotografií s využitím senzoru *Leap Motion*, abych umožnil uživatelům jednoduché a futuristické ovládání tohoto prohlížeče fotografií. Dalším důvodem byla možnost získat zkušenosti s vývojem aplikace řízené gesty, jelikož tento druh ovládání má velkou budoucnost a využití. Aplikace prohlížeče fotografií je navržena tak, aby byla co nejvíce uživatelsky přívětivá, intuitivní a umožňovala uživateli snadné ovládání. Aplikaci je také možno ovládat nejen s využitím senzoru *Leap Motion*, ale také klasickým způsobem tj. myší.

Následující kapitola 2 obsahuje soupis možností, které poskytuje ovládání gesty v současné době. Dále je zde uveden popis gesta, způsoby detekce gest a senzory, které jsou nejvíce rozšířeny. 3. kapitola se věnuje prohlížečům fotografií. Obsahuje informace o základních požadavcích, které jsou na tyto aplikace kladeny. V druhé části kapitoly jsou zmíněny některé prohlížeče, které využívají *Leap Motion*. Následující kapitola 4 popisuje informace o *Leap Motion SDK*. V 5. kapitole popisují jednotlivé kroky návrhu aplikace. Je zde rozepsán návrh a popis jednotlivých gest, funkcí aplikace, grafického rozhraní a nastavení. Kapitola 6. popisuje, jak byla aplikace vyvíjena a implementována. Dále strukturu programu a popis důležitých částí aplikace. Pro lepší zhodnocení a zpětnou vazbu je v 7. kapitole popsáno testování aplikace. Testování proběhlo formou dotazníku, a také pomocí přímého testování uživateli. Tato testování jsou zde vyhodnocena. V závěrečné kapitole 8 je zhodnocena celková aplikace. Zmíním zde i nápady a plány, jakým způsobem bych chtěl aplikaci v budoucnu rozšířit a vylepšit.

## Kapitola 2

# Detekce pohybu rukou a senzory pro detekci gest

Tato kapitola je zaměřena na všeobecné využití ovládání gesty v současné době. Dále se věnuje všeobecnému pojetí gest a možnostem detekce pohybu rukou s pomocí různých technologií. Jsou zde zmíněny i některé senzory pro snímání gest, které jsou v dnešní době dostupné a nejrozšířenější. Mezi nimi je i samotný senzor *Leap Motion*.

### 2.1 Současnost

Ovládání gesty zažívá v posledních několika letech velký boom. Dostáváme se do nové éry bezkontaktního ovládání, které umožňuje nové možnosti co se ovládání týče. Díky neustávajícímu technologickému pokroku disponují senzory a čidla vyšší přesností detekce pohybu a mají menší odezvu (rychlejší reakční čas). Postupně se také snižuje i jejich samotná cena a proto se zvyšuje i jejich dostupnost a celkové využití.

Nejvíce se v současné době využívá detekce pohybu především v herním průmyslu. Herní průmysl je velkým tahounem vývoje a šíření nových technologií tohoto typu. Herních konzolí, které lze ovládat s využitím gest se prodalo přes několik stovek milionů kusů<sup>[1]</sup>. Mezi tyto konzole patří především konzole Nintendo Wii, Xbox a PlayStation. S touto oblastí je také spojen i pojem virtuální reality, která je v současnosti velmi zmiňovaným tématem. Virtuální realita se do povědomí dostává především zásluhou projektu Oculus Rift<sup>1</sup>, který v roce 2014 koupila firma Facebook. V kombinaci s detekcí pohybu tak lze již dnes vytvořit přesvědčivé virtuální prostředí, s kterým lze pracovat pomocí gest a pohybů. Samotný Oculus Rift je velmi často využíván ve spojení se senzorem *Leap Motion*.

I když jsou herní konzole na prvním místě co se týče podílu na trhu, existují i jiná odvětví, ve kterých ovládání gesty nachází ještě větší uplatnění. Jedno z takových odvětví je medicína. V této sféře je kladen obrovský důraz na sterilitu. Právě sterilita je podle mého názoru nesmírně důležitým aspektem a výhodou ovládání pomocí gest. Je tak možné ovládání různých druhů zařízení na místech, kde jsou vysoké nároky na hygienu. Mezi takové místa patří různé laboratoře nebo operační sály. Tento druh produktů vytváří například firma TedCas<sup>2</sup>, přičemž využívá možnosti senzoru *Leap Motion*. Tyto produkty umožňují chirurgům na operačním sále prohlížení rentgenových snímků nebo jiných informací o pacientech bez nutnosti využití klasických periférií, které by v případě jejich použití způsobily

---

<sup>1</sup><https://www.oculus.com/>

<sup>2</sup><http://www.tedcas.com/en/products>

kontaminaci a nutnost opětovné dekontaminace rukou. Sterilního ovládání je dále možné využít u informačních kiosků na frekventovaných místech, jako jsou nádražní haly, letiště a jiné, tak aby se co možná nejvíce omezilo šíření různých virů a bakterií.

Velikou výhodou je také možnost pomoci lidem s některými druhy hendikepu. Pro hluchoněmé existuje zařízení UNI<sup>3</sup> od společnosti MotionSavvy. Toto zařízení je v podstatě tablet s integrovaným senzorem *Leap Motion*. UNI má dvě základní funkce a to rozpoznání znakové řeči právě díky *Leap Motionu* a její následné převedení do mluveného slova. Druhou funkcí je převod mluveného slova do textové podoby. S využitím tohoto zařízení mohou hluchoněmí lidé snadněji komunikovat s lidmi, kteří neumějí znakovou řeč a usnadnit jim tak jejich život.

V neposlední řadě je již dnes možné tímto způsobem ovládat například quadrokoptéry, RC modely atd. Je tedy pouze otázkou času, kdy se tento druh ovládání stane natolik spolehlivý a přesný, že se bude využívat pro ovládání dalších systémů. Lze tedy do budoucna předpokládat možnost ovládání například robotických zařízení. Tento druh ovládání proto čeká velká budoucnost.

## 2.2 Popis gesta

Před samotným popisem detekcí gest rukou, je třeba definovat co je to pojem gesto. Jak popisuje Liang [10], všeobecně jsou gesta brána jako pohyby rukou a těla, které mají předat určitou informaci. To je obecná definice, ale pro definici gesta z pohledu gest rukou je třeba si definovat co je to pozice ruky. Uvádím zde i anglickou definici pro lepší porozumění.

**Definice 1.** *Pozice ruky – Pozice ruky je její specifická poloha pozorovaná v určitou chvíli.*

**Anglická definice 1.** *Posture – A posture is a specific configuration of hand flexion observed at some time instance.*

Po samotné definici pozice ruky nyní lze snadno dodefinovat, co je bráno pod pojmem gesto.

**Definice 2.** *Gesto – Gesto je posloupnost po sobě jdoucích pozic ruky spojených pohyby během krátkého časového úseku.*

**Anglická definice 2.** *Gesture – A gesture is a sequence of postures connected by motions over a short time span.*

Podle Lamara [7] lze však tyto dvě definice rozdělit dále do dvou kategorií a to gesta statická a dynamická. Do statického gesta spadá pozice ruky, viz definice 1. Dynamického gesto je již samotné gesto, viz definice 2.

## 2.3 Metody detekce pohybu a gest

Pro možnost detekce pohybu je potřeba využít určitého druhu zařízení, které lze využít pro snímání a následné interpretování získaných informací. Uvádím zde rozdělení podle vstupních zařízení [13], která se v současnosti nejčastěji využívají. Každý typ vstupních zařízení se stále vyvíjí a díky technologickému pokroku se zlepšují jejich detekční vlastnosti.

---

<sup>3</sup><http://www.motionsavvy.com/#product>



## Vstupní zařízení

### Ovladačem řízené gesta:

Jedná se o elektronická zařízení, kterými lze detekovat pohyby a gesta. Mezi tato zařízení ovladačů patří například Wii Remote, Myo nebo Sony Move. Tato zařízení jsou popsána v podkapitole 2.4. Pro možnost snímání pohybu mají tyto ovladače v sobě integrovaný některý druh senzoru. Zmíním zde proto tři nejvyužívanější senzory, které se v ovladačích využívají. Díky MEMS<sup>4</sup> technologii lze tyto senzory vyrábět ve velmi malých velikostech v řádech milimetrů až mikronů. Takto vytvořená zařízení mohou dokonce obsahovat pohyblivé části.

- **akcelerometr**

Jedná se o elektromechanické zařízení, díky kterému lze měřit buď dynamické zrychlení objektu, které vzniká při pohybu zařízení, nebo také statické zrychlení, což je síla zemské gravitace na senzor [5].

- **gyroskop**

Elektromagnetické zařízení sloužící pro výpočet orientace a rotace zařízení. V dnešní době je již součástí většiny chytrých zařízeních pro určování směru náklonu [11].

- **magnetometr**

Senzor pro detekci a měření magnetického pole. Lze tak určit směr v závislosti na magnetickém poli Země [3].

### Hloubkové kamery

Hloubkové kamery umí detekovat vzdálenost objektu od kamery. Zmíním zde dva základní principy, které umožňují tuto detekci:

- **Strukturované světlo**

Pro aplikaci tohoto řešení je podstatná kamera společně se zdrojem strukturovaného světla. Nejčastěji je tímto zdrojem světla infračervený emitore(laser). Tento způsob detekce funguje na principu promítání známých světelných vzorů do prostoru. Pokud je osvětlen jakýkoli objekt, je tento vzor světla deformován. Z této deformace lze speciálními algoritmy dopočítat vzdálenost objektu od zařízení. [12]

- **ToF kamera**

*ToF* neboli *Time-of-flight* znamená v překladu “čas letu”. Jedná se konkrétně o rychlost světla. Existuje několik druhů kamer tohoto typu. Základní princip je však ten, že jsou do prostoru před kamerou vysílány světelné pulzy, které jsou případně odrazeny objekty před kamerou. Odražené světlo dopadá na senzor kamery s určitou časovou odezvou. Na základě znalosti této odezvy a konstantní rychlosti světla lze zjistit vzdálenost předmětu od kamery.

Velmi zajímavé využití této technologie je například projekt *Looking Around Corners*<sup>5</sup>, v překladu “Sledování za roh”. Popisuje možnost využití *ToF* kamery pro možnost pozorování objektů za rohem.

---

<sup>4</sup>Micro electro mechanical systems

<sup>5</sup><http://web.mit.edu/~velten/www/corner/>

## Kamera

Jednoduché snímání pomocí analýzy snímků poskytovaných kamerou. Jsou dostupné pouze 2D snímky, na kterých lze detekovat objekty. Tyto objekty jsou postupně sledovány. Na základně jejich pozice a tvaru lze v průběhu snímků zjistit jejich pohyb a popřípadě i gesta. Tohoto principu využívá senzor Sony Move.

## Stereo kamera

Jedná se o stejný princip detekce jako u předchozí varianty. Jsou zde však využity dvě kamery. To umožňuje lepší snímání objektů v prostoru, jejich přesnější pozici a vzdálenost od kamer. Tento princip je založen na principu lidských očí.

## 2.4 Senzory pro detekci pohybu a gest

Níže jsou uvedeny existující senzory, které lze využít pro snímání pohybu rukou. Každý z těchto senzorů se liší svou konstrukcí, způsobem detekce pohybu a jeho využitím. U jednotlivých zařízení zmíním jejich výhody i nevýhody. Zvolil jsem především takové senzory, které mají dostupné *SDK*<sup>6</sup>, což umožňuje vývojářům vytváření nových aplikací s využitím daného senzoru.

### Počítačová myš

Za jeden z prvních senzorů gest lze jistě považovat počítačovou myš, která byla vynalezena před více jak 50 lety. Myš je jednou z nejrozšířenějších počítačových periférií současnosti. I přes neustálý vývoj nových technologií zůstává koncept myši nezměněn. Myš umožňuje oproti dále zmíněným senzorům detekci pohybů a gest pouze ve 2D prostoru. I když se nejedná o technologickou novinku, chtěl jsem tuto periférii zmínit, protože se jedná o nejrozšířenější senzor pohybů a gest.

### Wii Remote

Společnost Nintendo je jedním z největších průkopníků co se týče ovládání gesty. Koncem roku 2006 vydala svou herní konzoli Nintendo Wii. Tato konzole byla první svého druhu a stala se velkým hitem mezi herními konzolemi. Konzoli je možné ovládat pomocí ovladače Wii Remote, neboli také Wiimote. Ovladač obsahuje mikroelektromechanický tříosý akcelerometr, díky kterému je možné snímat pohyby ovladače a ty následně přenést do konzole pro následné zpracování. Ovladač lze použít jako ukazatel na obrazovku televize. Tato funkce je možná díky infračervenému senzoru, kterým je vybaven samotný ovladač a panelu se sadou infračervených emitorů. Tento panel je třeba umístit na horní, nebo dolní okraj televize. Ovladač je po kalibraci schopen interpretovat pozici okraje televize a na základě této informace je možné zobrazit ukazatel přesně na pozici, kam ovladač ukazuje. Komunikace mezi konzolí a senzorem je prováděna pomocí bluetooth. Díky bluetooth je možné odesílat data i zpět do ovladače, který umožňuje i zvukovou nebo vibrační odezvu. U tohoto senzoru je nevýhodou nemožnost detekce pozice v prostoru. Nintendo ke svému senzoru poskytuje řadu možných doplňků a rozšíření. Jedním z těchto rozšíření je tzv. Nunchuk, který také obsahuje akcelerometr a po připojení k Wii Remote je možné snímat pohyby obou rukou simultánně [14].

---

<sup>6</sup>Software development kit

I přesto, že samotný ovladač má omezené možnosti, existuje SDK<sup>7</sup> pro vývoj v *C++*. Díky tomuto *SDK* tak získává data z ovladače přímo do počítače přes bluetooth. Díky *SDK* a infračervenému senzoru v ovladači je možné například implementovat elektronickou tabuli<sup>8</sup>, na kterou je možné kreslit pomocí infračerveného pera.



Obrázek 2.1: Wii Remote <sup>9</sup>

## Kinect

Kinect je pohybový senzor, který je primárně určen pro možnost ovládání herní konzole Xbox. Herní konzole společně se senzorem jsou vyvíjeny společností Microsoft. První verze Kinect 1.0 byla vydána v roce 2010 pro herní konzole Xbox 360. Koncem roku 2013 byla vydána novější a vyspělejší verze Kinect 2.0 pro konzoli Xbox One. Kinect 2.0 disponuje kamerou s rozlišením 1920x1080px oproti 640x480px u první verzi<sup>[2]</sup>. Další velmi podstatným rozdílem je způsob detekce hloubky u těchto senzorů. Novější verze disponuje lepší kvalitou detekce hloubky předmětů a využívá principu *ToF* kamery a starší verze technologii analýzy strukturovaného světla, viz 2.3.

Microsoft poskytuje vývojářům softwaru oficiální *SDK*<sup>10</sup> pro vývoj aplikací. Oficiální *SDK* podporuje pouze programovací jazyky *C++*, *C#*, *Visual Basic .NET*. *SDK* je však oficiálně dostupný pouze pro vývoj na platformě *Windows 7* a vyšších. Vzniklo proto několik alternativ *SDK*. Jednou z nejznámějších je open source projekt *OpenKinect*<sup>11</sup>, díky kterému lze vyvíjet nejen na platformě *Windows*, ale také na platformách *Linux* a *Mac*. *OpenKinect* dále umožňuje vývoj i v programovacích jazycích jako *Python*, *C/C++*, *C#*, *Actionscript*, *Java*, *Javascript* nebo *Lisp*. Kinect se díky svým možnostem využívá nejenom pro ovládání her, ale nachází uplatnění především pro vědecké projekty a taktéž pro možnosti využití v rozšířené realitě.

---

<sup>7</sup><http://www.btframework.com/wiimote.htm>

<sup>8</sup><http://www.uweschmidt.org/wiimote-whiteboard>

<sup>9</sup>Převzato z: <http://www.engadget.com/products/nintendo/wii/remote-controller/>

<sup>10</sup><https://www.microsoft.com/en-us/kinectforwindows/>

<sup>11</sup>[http://openkinect.org/wiki/Main\\_Page](http://openkinect.org/wiki/Main_Page)



Obrázek 2.2: Kinect One<sup>12</sup>

### PlayStation Move

PlayStation Move je zařízení od společnosti Sony, které slouží pro ovládání konzolí PlayStation. Zařízení bylo vydáno koncem roku 2010. Tento senzor je velmi podobný výše zmíněnému senzoru Wii Remote, ale oproti němu umožňuje detekci hloubkové pozice ovladače. I když tento senzor umožňuje detekci hloubky, pracuje zcela na jiném principu než Kinect. Uživatel musí pro ovládání her držet bezdrátovou “hůlku”, která pomocí bluetooth komunikuje s konzolí. Tato “hůlka” má na svém vrcholu plastovou RGB LED kouli, která může měnit svou barvu. Toho využívá kamera Sony Eye u konzole, která dokáže rozpoznat o který kontrolér se jedná. Díky barvě ovladače lze také určit jeho polohu a díky neměnné velikosti světelné koule lze s rychlou odezvou vypočítat vzdálenost ovladače od kamery. Tato poloha může být následně interpretována do herního prostředí. Ovladač má integrován tříosý lineární akcelerometr pro detekci pohybů, tříosý snímač úhlové rychlosti i magnetometr [15]. Existuje dostupné oficiální SDK Move.me<sup>13</sup>, nebo také open SDK<sup>14</sup>



Obrázek 2.3: PlayStation Move<sup>15</sup>

<sup>12</sup>Převzato z: <http://www.microsoftxboxone.cz/xbox-one-prislušenstvi/>

<sup>13</sup><https://code.google.com/p/moveme/>

<sup>14</sup><http://thp.io/2010/psmove/>

## Myo

Náramek od společnosti Thalmic patří do kategorie nositelné elektroniky. Tento náramek umožňuje snímáním elektrických biosignálů ve svalech ruky pomocí osmi EMG<sup>16</sup> nerezových snímačů lékařské kvality. Toto zařízení je určeno pro snadné ovládání prezentací, videí, a jiného multimediálního obsahu. Náramku trvá přibližně 5 minut “nahřívání” senzorů, než je možné jej využívat. To z důvodu ustalování elektrického spojení mezi zmíněnými senzory a svaly ruky. Před použitím senzoru je možné zkalibrovat senzor pro aktuálního uživatele a na základě kalibrace vytvořit jeho osobní profil. Mimo EMG senzory obsahuje náramek také devítiosý IMU<sup>17</sup> senzor, který se skládá z tříosého gyroskopu, tříosého akcelerometru a tříosého magnetometru. Data z těchto senzorů zpracovává procesor *ARM Cortex M4*. Ovladač je podle výrobce schopen pracovat na jedno nabití celý den. Zařízení poskytuje základní gesta sevření v pěst, mávnutí vpravo/vlevo, roztažení prstů a dvojité poklikání. Senzor je možné propojit s platformami *Windows*, *Mac*, *iOS* a *Android* pomocí bluetooth [6]. Existuje již mnoho aplikací, které náramek využívají. Tyto aplikace jsou dostupné na oficiálních stránkách Myo Marketu<sup>18</sup>. Je dostupné SDK<sup>19</sup> pro vývoj v *C++* popřípadě open source API<sup>20</sup>, které umožňuje vývoj v *Pythonu* na platformě *Linux*. Myo náramek se v současnosti řadí mezi ty nejvyspělejší senzory.



Obrázek 2.4: Myo náramek [6]

## Leap Motion

*Leap Motion* je zařízení od stejnojmenné firmy, která vznikla v roce 2010. Zařízení *Leap Motion* pracuje na principu optických senzorů a infračerveného světla. Tento senzor se vyznačuje velmi vysokou přesností určování pozice objektů a rychlou odezvou. Vývojářská verze senzoru má malé rozměry a to 3” x 1” x 0.5” o hmotnosti 45 gramů. Tato malá velikost senzoru je však kompenzována celkovým zpracováním surových (nezpracovaných) dat, které jsou přenášena do počítače. Tyto data jsou analyzována a přepočítávána až v počítači

<sup>15</sup>Převzato z: <http://www.amazon.com/Playstation-Move-Motion-Controller-3/dp/B002IOJ5IU>

<sup>16</sup>Elektromyografie

<sup>17</sup>Inertial measurement unit

<sup>18</sup><https://market.myo.com/>

<sup>19</sup><https://developer.thalmic.com/>

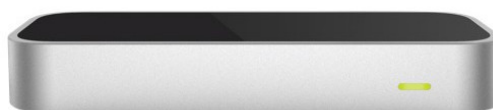
<sup>20</sup><https://github.com/f825f5242ed81a32cd04e5269665f40a/libmyolinux>

pomocí speciálních složitých algoritmů. To má za následek vyšší nároky na technickou výbavu počítače. V tabulce 2.1 jsou vypsané minimální hardwarové požadavky senzoru.

System	<i>Windows</i> ©7/8, <i>Mac</i> ©OS X 10.7 or <i>Linux</i>
Processor	AMD Phenom™ II or Intel©Core™ i3/i5/i7 processor
RAM	2 GB

Tabulka 2.1: Minimální systémové požadavky

Nejpodstatnějšími částmi senzoru jsou dvě kamery se širokoúhlými objektivy a trojice infračervených světél, které osvětlují snímaný prostor. Senzor má velké zorné pole. To je konkrétně 150° s možností sledovat objekty ve vzdálenosti od 25 až do 600 milimetrů. Vzdálenost je limitována především výkonem infračervených diod osvětlují prostor nad senzorem, které jsou závislé na možnostech rozhraní *USB* [9]. Pro senzor jsou dostupné desítky placených i neplacených aplikací různých žánrů, které si mohou uživatelé stáhnout a vyzkoušet. Jedná se především o grafické aplikace, hry, výukové programy a aplikace umožňující rozšířené ovládání počítače. Tyto aplikace se nacházejí na oficiálních stránkách internetového obchodu<sup>21</sup>. *SDK* pro vývoj aplikací využívajících *Leap Motion* je podrobněji popsáno v kapitole 4. V poslední době se objevuje čím dál více projektů využívajících *Leap Motion*.



Obrázek 2.5: Leap Motion [9]

---

<sup>21</sup><https://apps.leapmotion.com/>

## Kapitola 3

# Prohlížeče fotografií

V současnosti existuje mnoho aplikací prohlížečů fotek a obrázků, které se liší svojí funkcionalitou a především kvalitou celkového zpracování. Tato kapitola popisuje tento typ aplikací, jejich vlastnosti a rozdíly mezi nimi. V podkapitole 3.2 jsou vypsány některé již existující fotogalerie, které je možné ovládat pomocí senzoru *Leap Motion*.

### 3.1 Rozdělení a popis prohlížečů obrázků

Prohlížečů obrázků existuje nepřehledné množství, liší se však svými funkcemi. Již z názvu je jasné, že hlavním cílem těchto aplikací je možnost prohlížení obrázků nebo fotek. Důležité je zmínit, že v textu jsou brány obrázky a fotografie jako totožný pojem. A to z toho důvodu, že se v obou případech jedná o grafické soubory, které spadají do množiny grafických datových formátů. Zvolil jsem dva typy těchto programů, které považuji za důležité popsat před popisem mnou navržené a implementované aplikace. Níže jsou tedy uvedeny vlastnosti a požadavky na prohlížeče obrázků a fotogalerie.

#### Prohlížeč obrázků

Funkcí tohoto typu aplikací je především zobrazovat obrázky, popřípadě obrázky procházet v rámci jedné složky, což je jejich hlavním účelem. V neposlední řadě mohou poskytovat základní editaci obrázků.

#### Základní funkce prohlížečů obrázků:

- prohlížení obrázků
- rotace, přiblížení a oddálení obrázků
- ořezání obrázků
- prohlížení v celoobrazovkovém módu
- prezentace
- zobrazení náhledů obrázků ve zvolené složce
- tisknutí obrázků

## Fotogalerie

Fotogalerie jsou v podstatě rozšířené prohlížeče obrázků. Hlavním rozdílem a výhodou fotogalerií je možnost vytváření kolekcí fotografií. Ty lze dále členit do přehledných galerií, které je možné seskupovat třeba podle data. Jelikož galerie pracuje především s fotografiemi, umožňuje také jejich editaci a to například: úprava expozice, barevné korekce, redukce červených očí, změna kontrastu, zostření, změna kompozice, redukce šumu a jiné.

### Základní funkce fotogalerií:

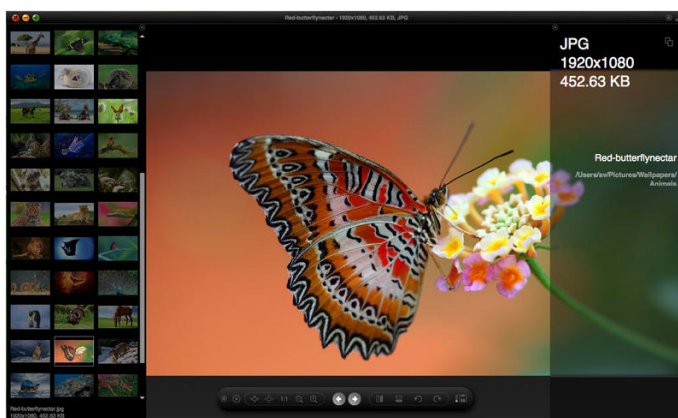
- organizace fotografií a značení jednotlivých fotek
- zobrazení informací *EXIF*<sup>1</sup>
- vyhledávání
- propojení s kamerou
- sdílení fotek
- editace a úprava fotografií

## 3.2 Prohlížeče obrázků ovládané gesty

Níže jsou uvedeny aplikace pro prohlížení obrázků, které je možné ovládat pomocí zařízení *Leap Motion*. Ještě před popisem těchto aplikací bych zde chtěl zmínit to, že ovládání pomocí *Leap Motion* by bylo možné využít u jakékoliv již existující aplikace pro prohlížení obrázků v případě implementování patřičného rozhraní.

### Image Viewer Deluxe

Jedná se o velmi kvalitně zpracovaný prohlížeč fotografií s přehledným grafickým rozhraním. Nevýhoda této aplikace je především to, že není multiplatformní. Je dostupná pouze na platformě *Mac OS*. Tato aplikace umožňuje snadné otevírání složek a obrázků pouhým přetažením na okno aplikace. Dalšími možnostmi jsou rotace, přetočení obrázků a možnost editovatelného uživatelského prostředí. Lze zobrazovat i *EXIF* informace [4].



Obrázek 3.1: Image Viewer Deluxe [4]

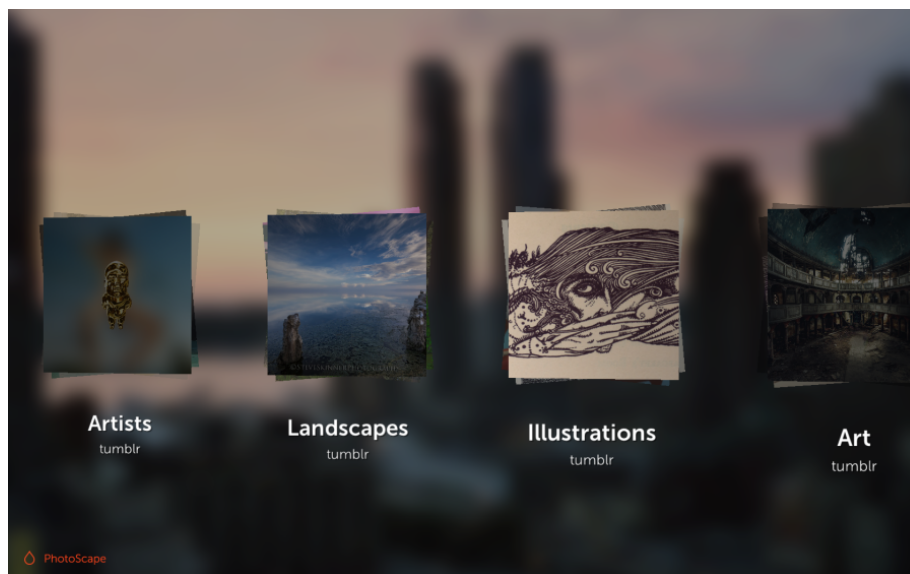
---

<sup>1</sup>Exchangeable image file format



## PhotoScape

Tato aplikace počítá primárně s využitím *Leap Motionu*. Lze ji tak ovládat zcela pomocí gest. Jedná se především o skvěle graficky propracovanou aplikaci, která umožňuje prohlížení napodobující 3D projekci, což při ovládání *Leap Motionem* umožňuje realistický vjem. Galerie podporuje několik módů prohlížení obrázků a to: Grid View, Belt View a Timemachine. Další výhodou je možnost přímého prohlížení obsahu na některých internetových službách jako je Instagram, Flickr a 500px. Nevýhodou aplikace je opět především její dostupnost pouze pro platformu *Mac OS 10.7* a vyšší [8].



Obrázek 3.2: PhotoScape [8]

## Kapitola 4

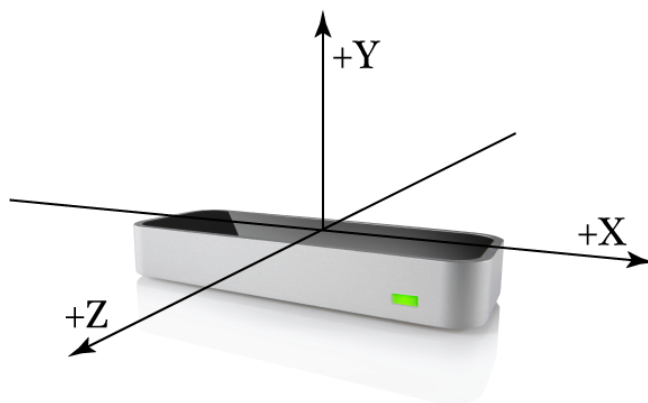
# Leap Motion SDK

Tato kapitola se zabývá *API*<sup>1</sup> pro vývoj programů s využitím *Leap Motion*. Obsahuje základní popis *Leap Motion API* a to, jakým způsobem lze získávat data o aktuálním dění nad senzorem. Dále obsahuje popis nejdůležitějších tříd *API*. Veškeré informace jsou získány přímo z oficiálních stránek dokumentace *API*[9].

### 4.1 Leap Motion SDK

Společnost *Leap Motion*, jak už jsem zmínil v popisu senzoru poskytuje velkou podporu programátorům v podobě kvalitně zpracovaného a zdokumentovaného *API*, pro možnost vývoje aplikací využívajících *Leap Motion*. *SDK* je dostupné pro všechny hlavní platformy jako je *Linux*, *OSX* a *Windows*. *SDK* je dostupné pro programovací jazyky *JavaScript*, *Unity*, *C#*, *C++*, *Java Python*, *Objective-C* a *Unreal*. Jelikož existuje podpora pro více programovacích jazyků, zvolil jsem *API* pro programovací jazyk *Java*, ve kterém je vyvíjena aplikace. V těchto *API* může být menší rozdíl, ale základní princip a získávání dat je stejný.

Senzor dokáže rozpoznat objekty rukou, prstů a nástrojů, které se podobají prstům. Souřadnice objektů jsou udávány v milimetrech a to s vysokou přesností. Důležité je zmínit, že pozice objektů jsou udávány v pravotočivé soustavě prostorových kartézských souřadnic. Dále jsou uvedeny nejpodstatnější třídy *API* a jejich základní popis.



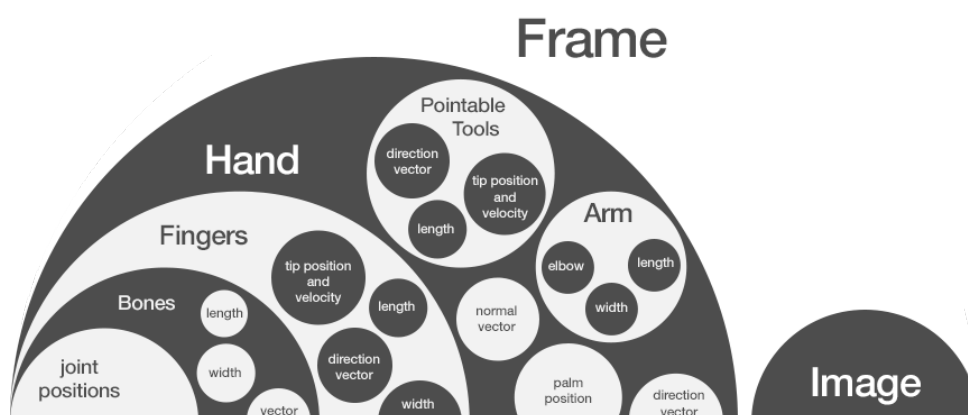
Obrázek 4.1: Leap Motion – osy [9]

---

<sup>1</sup>Application programming interface

## Frame

*Frame*, neboli snímek je základní fragment, který poskytuje *API* senzoru. Každý *Frame* obsahuje veškeré informace o objektech rukou, prstů a nástrojích, které se v daný okamžik nacházejí nad senzorem. Na obrázku 4.2 je přehledně zobrazena struktura objektů nacházejících se v každém snímku. Z tohoto obrázku také jasně vyplývá, že *Frame* je kořen celého datového modelu. Každý *Frame* je jednoznačně identifikovaný svým *id*. Dále pak každý snímek obsahuje informaci o aktuálním počtu snímků za sekundu (*FPS*<sup>2</sup>) a časové známce snímku. *Frame* je vždy jednoznačně identifikovatelný pomocí svého *id*, které se inkrementuje každým novým snímkem (někdy může být přeskočen, nebo vynechán kvůli omezeným zdrojům počítače).



Obrázek 4.2: Leap Motion API struktura [9]

## Controller

Jedná se o hlavní třídu rozhraní pro komunikaci se samotným *Leap Motionem*. Pomocí tohoto objektu lze získávat nová data o aktuálním dění nad senzorem a to ve formě snímků *Frame*. Dále lze pomocí *Controlleru* měnit základní konfiguraci senzoru. Získávání nejaktuálnějších snímků je umožněno pomocí dvou způsobů. První z nich je tzv. *polling*, což je forma periodického dotazování na *Controller* zda jsou dostupné nové snímky pro zpracování. Tato metoda má výhodu v tom, že aplikace má možnost načítat nová data až po zpracování veškerých získaných dat. Druhým možným způsobem získání dat je s využitím tzv. *Callbacků*, kdy je instance třídy *Listener* informována o nových událostech. *Controller* obsahuje buffer, díky kterému lze přistupovat až k 60ti snímkům zpět od aktuálního snímku i informace o všech připojených senzorech.

## Listener

Tato třída definuje metody, které jsou vyvolány v případě využití *Callbacků* od *Controlleru*. Tyto metody je možné překrýt vlastní definicí metody. Jedná se především o metodu *on-Frame()*, která je volána při novém *Framu Controllerem*. Další metody umožňují například vyvolat akci při připojení senzoru, jeho odpojení atd.

<sup>2</sup>Frame per second

## ***Hand***

Třída *Hand*, neboli ruka poskytuje informace o ruce, jako je její identifikační *id*, aktuální pozice, směru, typu, síle stisku a dalších. Každý objekt *Hand* obsahuje seznam jí přiřazeným prstů.

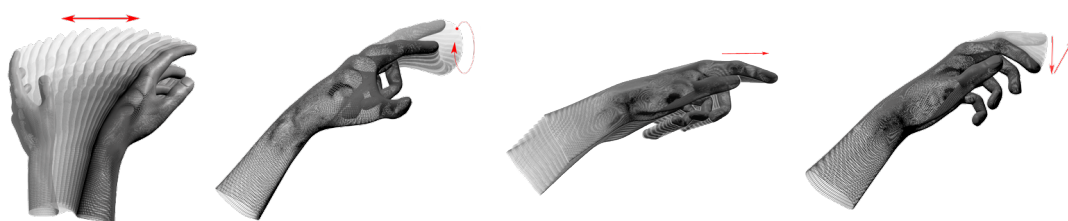
## ***Finger***

*Finger* je typ objektu, který definuje prst. Každý prst je přiřazen právě jednomu objektu *Hand*. *Finger* je definován vlastním *id* a obsahuje informace o aktuální pozici, směru, rychlosti, délce, šířce a především také o typu konkrétního prstu.

## ***Gesture***

*API* přímo poskytuje detekci základních gest. Seznam gest lze získat z *Framu*. Tak jako ostatní objekty i gesto má své unikátní *id*. Každé z těchto gest má svůj typ. Typy gest jsou uvedeny níže a také na obrázku 4.3.

- *Swipe* – Gesto mávnutí nad senzorem.
- *Circle* – Jedná se o gesto kruhu provedeného nad senzorem.
- *Screen Tap* – Gesto “ťuknutí” do prostoru nad senzorem.
- *Key tap* – Gesto podobné zmáčknutí klávesy v prostoru nad senzorem.



Obrázek 4.3: Leap Motion API – gesta [9]

# Kapitola 5

## Návrh

V této kapitole popisuji návrh aplikace a ty nejdůležitější části návrhu prohlížeče. Některé problémy návrhu byly zjištěny až v průběhu implementace, proto byl návrh průběžně upravován až do této finální podoby. Je zde popsán výsledný návrh aplikace, jednotlivých gest, funkcí, ovládání a nastavení.

### 5.1 Popis návrhu

Při návrhu jsem se snažil navrhnout přehlednou aplikaci prohlížeče fotografií, která umožní uživatelům jednoduché ovládání pomocí gest. Ve [3.](#) jsem popisoval základní požadavky na prohlížeče obrázků. Těchto požadavků jsem se snažil držet. Aplikace tedy bude podporovat tyto funkce procházení, rotaci a přibližování obrázků. Dále také prezentační mód, možnost smazání, odsmazání obrázků a nastavení aplikace.

Při finálním návrhu byly brány v potaz především možnosti a výhody programovacího jazyka *Java* a platformy *JavaFX*. Právě s využitím *JavaFX* byli navrženy animace, které poskytují aplikaci příjemnější ovládání a zlepšují celkový dojem. Návrh v *Javě* počítal s multiplatformní aplikací pro platformy *Linux* a *Windows* popřípadě *Mac*. Aplikace má za cíl podporovat především základní formáty typu *PNG*, *JPG* a také možnost zobrazení animací *GIF*. Při počátečním návrhu jsem se inspiroval již zmíněnými aplikacemi ovládaných gesty a to především galerií *Image Viewer Deluxe*, o které jsem se zmínil v podkapitole [3.2.](#)

### 5.2 Návrh funkcí a gest

Při návrhu gest jsem se snažil co nejvíce využít možností poskytovaných *Leap Motion API*. Většina gest je navržena tak, aby byla jednoduše zapamatovatelná a co možná nejintuitivnější. Pro lepší odezvu uživateli jsem navrhl zobrazení polohy ruky v okně aplikace. Níže jsou popsány všechny navržené funkce aplikace. U každé funkce je uveden popis toho, jakou akci provádí a také gesto, kterým je daná funkce vyvolána. Pro lepší názornost je u každé funkce vložen obrázek<sup>1</sup>, který zobrazuje gesto potřebné k vyvolání funkce.

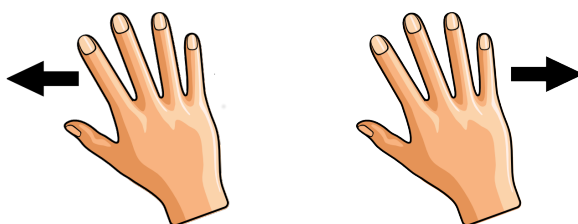
---

<sup>1</sup>Převzaty a upraveny z: <https://github.com/mariuszostrowski/gesty>

## Seznam funkcí podporovaných aplikací:

### Následující/předchozí obrázek

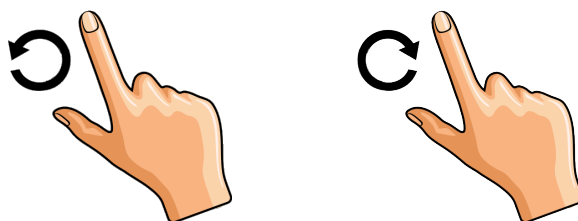
Tato funkce je jedna z nejpodstatnějších vlastností galerie. Jedná se o přechod mezi jednotlivými obrázky, jak je zobrazeno na obrázku 5.1. K provedení této akce je třeba mávnout rukou nad senzorem, a to buď směrem doleva nebo doprava. Na základě směru gesta se provede přepnutí na následující nebo předchozí obrázek. Senzor snímá rychlost pohybů a v případě mávnutí rychlejšího jak 1000 mm/s je detekováno toto gesto.



Obrázek 5.1: Gesto – Následující a předchozí obrázek

### Rotace

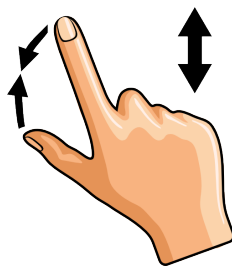
Toto gesto slouží pro otočení obrázku o 90° na základě směru rotace gesta. Obrázky, které mají vyšší rozlišení než jsou rozměry okna prohlížeče jsou roztaženy tak, aby se využil prostor okna aplikace co nejvíce a poměr stran obrázku zůstal nezměněn.



Obrázek 5.2: Gesto – Přiblížení a oddálení

### Přiblížení/oddálení

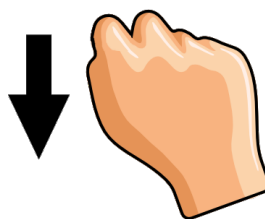
Pro přiblížení obrázku bylo navrženo dvoukrokové gesto tak, aby bylo omezeno náhodné spuštění této akce. Pro spuštění gesta je proto zapotřebí nejprve spojit palec a jakýkoliv jiný prst. Nyní se sepne přibližovací mód. Od této chvíle je snímána z-ová souřadnice ruky. Na základě této souřadnice je zobrazený obrázek buď přibližován nebo oddalován. Všechny ostatní gesta jsou během tohoto módu deaktivovány a funguje pouze tato jedna funkce.



Obrázek 5.3: Gesto – Levá a pravá rotace

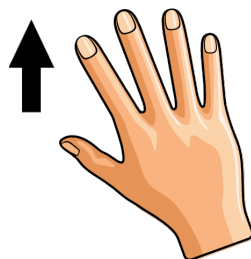
### Smazání/odsmazání

Funkce smazání je potencionálně velmi kritickou operací, jelikož obrázky a fotografie jsou poměrně důležité soubory, které nelze okamžitě po detekci gesta smazat. Může dojít jak k chybné detekci samotného gesta, nebo popřípadě nechtěným provedením gesta uživatelem. Z tohoto důvodu je toto gesto navrženo podobně jako předchozí gesto. Uživatel musí provést právě dva kroky pro jeho potvrzení gesta. Gesto funguje na principu *drag and drop*<sup>2</sup> a to tak, že je zahájeno sevřením ruky v pěst. Od tohoto může být obrázek přetažen na spodní panel, který indikuje mazací mód. Uvolněním ruky je gesto buď potvrzeno, nebo zrušeno v závislosti na *y*-ové souřadnici ruky. Gesto se snažím popsat obrázkem 5.4.



Obrázek 5.4: Gesto - Smazání obrázku

V případě, že je detekováno gesto pro smazání obrázku, je obrázek pouze označen jako tzv. předsmazaný. To znamená, že takto ‘smazaný’ obrázek lze opětovně odsmazat za pomoci gesta zobrazeného na obrázku 5.5. K úplnému smazání souborů dojde až při ukončení aplikace a potvrzení uživatelem, že si smazání zvolených obrázků opravdu přeje.



Obrázek 5.5: Gesto – Odsmazání obrázku

---

<sup>2</sup>táhni a pusť

## Fullscreen

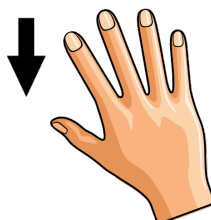
Pro možnost prohlížení fotografií na celé obrazovce, může uživatel dvojitým kliknutím do prostoru přepnout aplikaci na celou obrazovku.



Obrázek 5.6: Gesto – Fullscreen

## Prezentace

Předposlední užitečnou funkcí aplikace je mód prezentace, který po spuštění automaticky přepíná snímky a prochází postupně všechny obrázky. Rychlost přepínání obrázků lze jednoduše upravit v nastavení aplikace.



Obrázek 5.7: Gesto – Prezentace

## Ořezání

Nakonec uvádím navrženou funkci, která nebyla v konečné aplikaci implementována. Tato funkce slouží pro možnost ořezání obrázku.

## 5.3 Grafické rozhraní

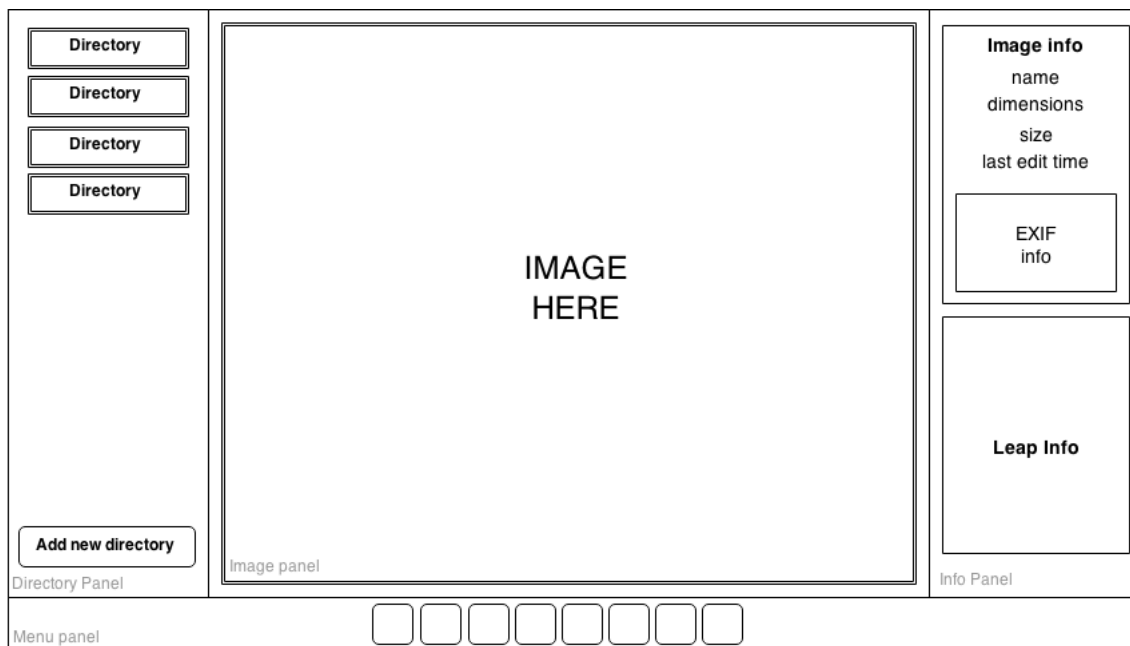
Uživatelské rozhraní aplikace bylo navrženo tak, aby uživatel mohl intuitivně ovládat aplikaci i bez předchozích zkušeností. *GUI* bylo rozvrženo na čtyři základní panely. Těmito panely jsou *Menu panel*, *Info panel*, *Folder panel*, a *Image panel*. Základní rozvržení aplikace je zobrazeno na obrázku 5.8.

### Navrhované panely:

Na spodní části je zobrazen *Menu panel*, který je důležitý především pro ovládání aplikace myší. Obsahuje ikony pro spuštění funkcí, které jsou popsány v předchozí kapitole. U tohoto panelu je možné nastavit skrývání po odjetí myši mimo tento panel. Dalším panelem je *Info panel*. Na tomto panelu jsou uvedeny důležité informace o právě zobrazeném obrázku, jako je



název souboru obrázku, rozlišení, velikost a datum poslední editace. Tento panel obsahuje i index zobrazeného obrázku v aktuálně zvoleném adresáři. *Folder panel* je důležitým panelem pro možnost přidávání složek. Je zde uveden seznam složek s obrázky. Tyto složky si může sám uživatel navolit. Při načtení aplikace je pak načtena vždy jako první právě první složka z tohoto seznamu. Posledním navrženým panelem je *Image panel* pro zobrazování obrázků. Již v návrhu jsem počítal s využitím animací pro přechody snímků a skrývání jednotlivých panelů.



Obrázek 5.8: Prohlížeč obrázků – návrh grafického rozhraní

## 5.4 Nastavení

Při návrhu bylo samozřejmé počítat s tím, že bude aplikace spouštěna opakovaně. Proto bylo nutné navrhnout způsob ukládání nastavení aplikace. Nastavení bylo navrženo tak, aby mohl uživatel upravit především vzhled a chování aplikace podle svých představ. Dále bude možné nastavit vzhled aplikace barvy a průhlednost panelů, rychlosti animací a jiné.

## Kapitola 6

# Implementace

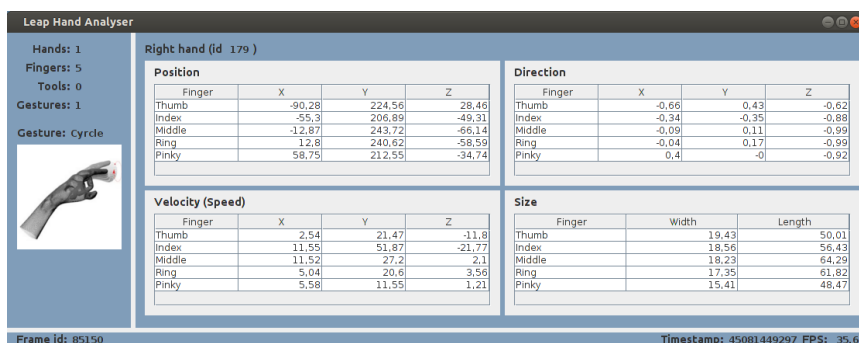
V této kapitole jsou popsány nejdůležitější kroky vývoje samotné aplikace. Je zde zmíněn počáteční vývoj jednoduchého analyzátoru dat ze senzoru. Dále tato kapitola obsahuje popis struktury kódu a popis některých nejpodstatnějších tříd projektu. V podkapitole grafického rozhraní 6.6 je uveden popis implementace *GUI*. Tato podkapitola obsahuje také snímky výsledné aplikace, kterou se podařilo vytvořit.

### 6.1 Popis celkového vývoje

Aplikace je implementována v programovacím jazyce *Java*. Vývojové prostředí jsem zvolil *Eclipse Luna* a to v Linuxové distribuci *Ubuntu 14.04.2 LTS*.

Ještě před implementací samotného prohlížeče fotografií byla vytvořena jednoduchá aplikace pro prezentaci získaných dat ze senzoru (obr. 6.1). Sloužila především pro lepší pochopení práce s *Leap Motion API*. Pro implementaci tohoto analyzátoru jsem použil *Java* knihovnu *Swing*. Při implementování této aplikace jsem zjistil omezení, která sebou přináší knihovna *Swing*, proto jsem hledal jiné alternativy. Zjistil jsem, že knihovna *Swing* je již nějakou dobu zastaralá a byla nahrazena novou platformou *JavaFX*.

Při celkové implementaci aplikace jsem se snažil dodržovat správnou objektovou strukturu kódu a jeho přehlednost. Také byly využity některé návrhové vzory, jako jsou například prostředník a jedináček. Drobnějším problémem při implementaci byl vývoj na méně výkonném počítači, což společně s vyššími nároky senzoru *Leap Motion* na procesor mělo za následek občasné sekání aplikace a především animací.



Obrázek 6.1: Leap Motion analyzátor

## 6.2 Struktura projektu

Pro lepší přehlednost jsem rozdělil třídy projektu do jednotlivých balíčků podle jejich funkce a využití. Dále jsou popsány jednotlivé balíčky a jejich účel. V každém balíčku jsou roze-psány některé nejpodstatnější dané třídy a jejich primární účel.

### gallery

Balíček obsahuje hlavní třídy projektu, které jsou často využívány.

- *Gallery* - hlavní třída programu
- *Globals* - třída obsahuje *static* proměnné, které využívají jednotlivé objekty programu
- *Connector* - je třída implementována jako jedináček, která usnadňuje komunikaci mezi některými objekty aplikace
- *LeapListener* - třída implementující zpracování gest
- *EditedInfo* - třída umožňující ukládání informací o změnách obrázku

### handlers

V tomto balíčku jsou implementováni tzv. správci, kteří se starají o práci se soubory, s gesty a s nastavením.

- *FileHandler* - jedná se o třídu, která se stará o celkovou správu souborů
- *SettingsHandler* - stará se o načítání i ukládání nastavení
- *GestureHandler* - zpracovává požadavky od *LeapListeneru* o provedených gestech a změnách pozic rukou

### panels

Tento balíček obsahuje třídy s jednotlivými panely, ze kterých se skládá celé *GUI* aplikace. Celkem obsahuje 9 panelů a hlavní scénu.

- *GalleryPane* - hlavní scéna aplikace, která obsahuje níže zmíněné panely
- *ImagePanel* - každý zobrazený obrázek je instancí této třídy
- *MenuPanel* - menu panel umožňující ovládat aplikaci myší
- *InfoPanel* - informační panel zobrazující informace o zobrazeném obrázku
- *FolderPanel* - panel se složkami obrázků
- *NotificationPanel* - notifikační panel zobrazující provedené akce
- *SavePanel* - panel zobrazující změny
- *DeletePanel* - panel objevující se v případě mazání obrázku
- *DrawPanel* - vykreslovací panel ukazatele prstu

## labels

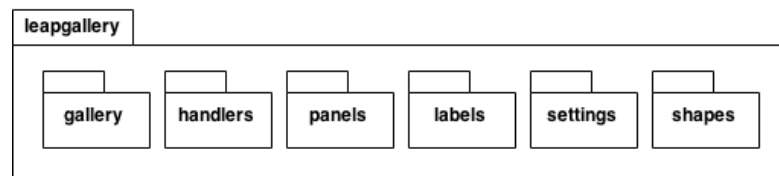
Balíček *labels* obsahuje prvky pro snadnou jednotnou úpravu tlačítek grafického rozhraní aplikace.

## settings

Jedná se o velmi rozsáhlý balíček co se týče celkového počtu tříd. Tento balíček obsahuje veškeré potřebné třídy pro zobrazení nastavení. Jsou zde implementovány všechny podkategorie nastavení a také komponenty pro možné změny v nastavení.

## shapes

Obsahuje třídy popisující tvary vykreslující se nad zobrazeným obrázkem. Jedná se například o třídu *RightCircle*, což je objekt tvaru kruhu, který slouží jako ukazatel aktuální pozice výchozího prstu nad senzorem.



Obrázek 6.2: Struktura balíčků

## 6.3 Gesta

Správu samotných gest má na starosti instance objektu *LeapListener*, ve kterém je implementováno zpracování veškerých dat získaných ze senzoru jako jsou gesta a pozice rukou. Aplikace zpracovává nové snímky vždy, když jsou dostupné. Využívá *GestureHandler*, který je určen jako rozhraní mezi *LeapListenerem* a zbytkem aplikace. *GestureHandler* umožňuje rychlou úpravu chování při konkrétních gestech. Přes toto rozhraní se předávají informace o provedených gestech a aktuální pozici výchozího prstu nad senzorem. To umožňuje vykreslovat na scénu aplikace ukazatel prstu ve tvaru kruhu.

## 6.4 Nastavení

Nastavení je implementováno pro možnost ukládání vlastní konfigurace uživatelem. Při spuštění aplikace se provede kontrola, zda již existuje soubor s nastavením v domovském adresáři uživatele. Pokud ne, je vytvořen nový skrytý adresář s konfiguračním souborem aplikace. V případě, že již konfigurační soubor existuje, načtou se z něj data o nastavení, která jsou nahrána do proměnných v třídě *Global*. Z této třídy mohou načítat konfigurační proměnné jednotlivé objekty dle potřeby. Konfigurační soubor obsahuje dvojice hodnot a to název proměnné a uloženou hodnotu. V případě neexistence určité proměnné v konfiguračním souboru, je příslušné proměnné v třídě *Global* přiřazena výchozí hodnota.

## 6.5 Ukládání změn

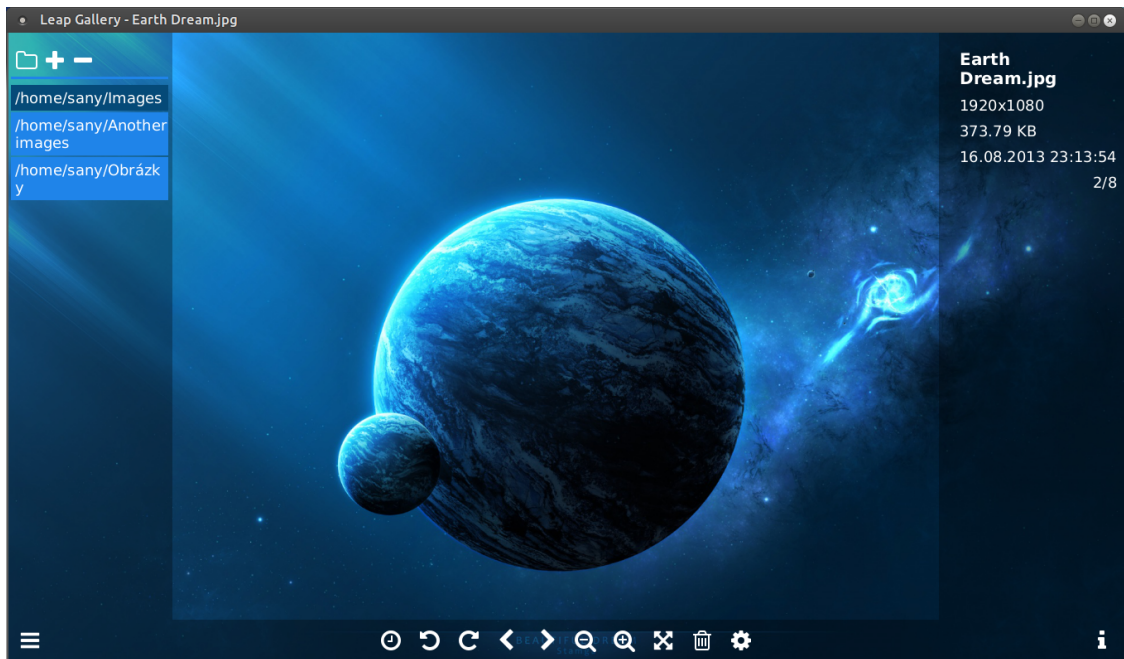
Jakákoliv změna, jako rotace a smazání obrázku, je ukládána pomocí *EditedInfo*. Před ukončením programu je ověřeno, zda uživatel provedl některou ze změn. V případě, že změna byla provedena, je uživatel dotázán pomocí vyskakovacího modálního okna, zda si opravdu přeje dané změny provést nebo zrušit. Toto okno obsahuje seznam názvů obrázků a informaci o tom, co bude v případě potvrzení změn s daným obrázkem provedeno. Pokud je zvolena možnost aplikovat změny, je s nimi provedena příslušná akce a aplikace je následně ukončena. Při implementaci jsem zjistil, že nedostatkem aplikace je operace rotace a následné uložení obrázku. Uložení obrázku je provedeno pomocí ztrátového převodu do *JPG*. To má za následek menší výsledný soubor a také o něco menší kvalitu výsledného obrázku.

## 6.6 Grafické rozhraní

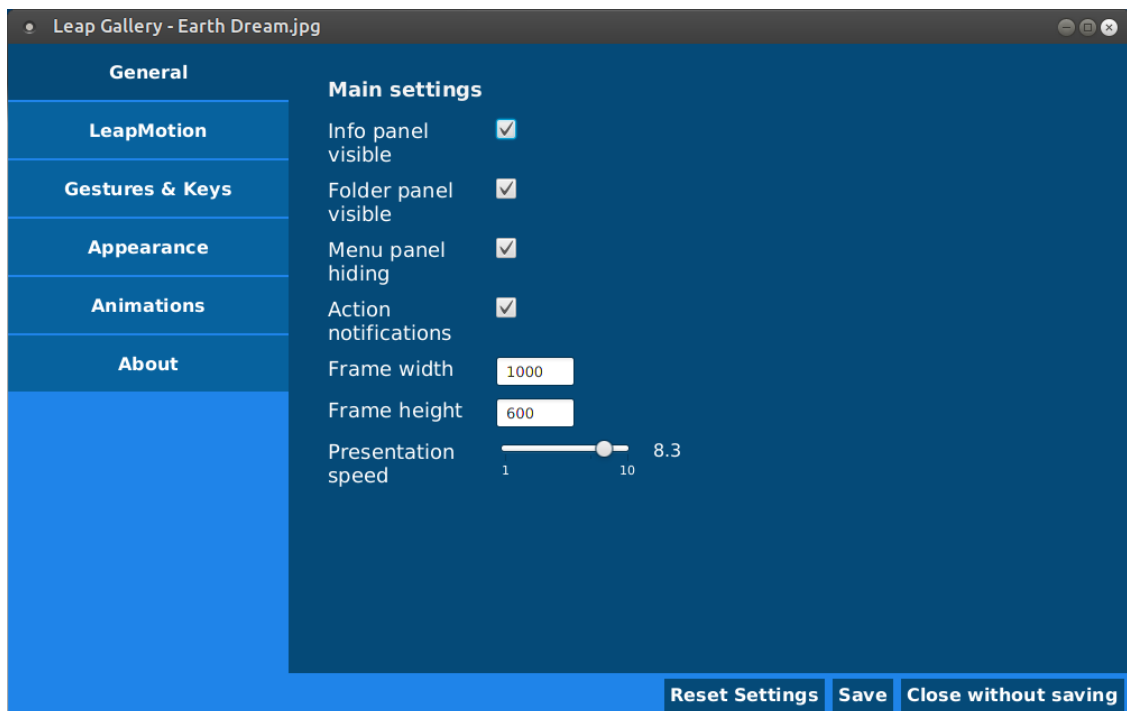
Implementace grafického rozhraní se skládala především ze správného pozicování jednotlivých komponent a panelů v *GUI*. Při jeho implementaci jsem se snažil vycházet z již zmíněného návrhu grafického rozhraní v kapitole 3. Základní prvky grafického rozhraní se tedy skládají z *MenuPanelu*, který obsahuje ikony<sup>1</sup> sloužící k ovládání aplikace myší. Dále také z *FolderPanelu*, který umožňuje uživateli přidávat nové adresáře s obrázky a zároveň obsahuje seznam již zvolených adresářů. Důležitým panelem je *InfoPanel* zobrazující informace o aktuálním obrázku. Pro zpětnou odezvu o provedených gestech a akcích uživateli byl přidán *NotificationPanel*, který slouží pro zobrazení průběžných informací o provedených akcích. Panely skýtají možnost skrytí tak, aby bylo umožněno především nerušené prohlížení obrázků a fotek. Důležitou částí při implementaci grafického rozhraní jsou také animace, které aplikaci přidávají lepší celkový dojem. Dále byly v *GUI* zakomponovány průhledné prvky. Jako barevné schéma aplikace bylo použito spíše tmavější prostředí v kombinaci s odstíny modré, která se vyskytuje v ovládacích prvcích a také v nastavení.

---

<sup>1</sup><http://fortawesome.github.io/Font-Awesome/3.2.1/>



Obrázek 6.3: Leap Gallery aplikace



Obrázek 6.4: Leap Gallery nastavení

# Kapitola 7

## Testování

Celkové testování aplikace proběhlo ve dvou fázích a to formou dotazníku i formou přímého testování aplikace uživateli. Jelikož se jedná o úplně rozdílný typ testování, vyhodnotil jsem dané testy zvlášť.

### 7.1 Hodnocení dotazníkem

Vytvořil jsem online formulář, který obsahoval otázky týkající se gest zvolených pro ovládání aplikace, dále celkového grafického rozhraní a pohledu hodnotících na technologie bezdotykového ovládání. K formuláři byly přiloženy ukázkové snímky aplikace, obrázek s přehledem dostupných gest a prezentační video zobrazující všechny funkce aplikace i s akcemi ruky. Dotazník vyplnilo celkem 23 dotázaných. Věkový rozptyl hodnotících byl mezi 18 až 71 lety a průměrný věk byl 27 let. Níže jsou uvedeny jednotlivé otázky a odpovědi na ně.

**Otázka: Je aplikace přehledná?**

Na tuto otázku mohli hodnotící odpovídat klasickým známkovacím způsobem jako ve škole. Průměrná výsledná známka toho, jak se hodnotícím zdála být aplikace přehledná vyšla 1.57. Podle tohoto hodnocení lze usoudit, že ovládací prvky a jednotlivé panely jsou přehledné.

**Otázka: Jak se vám líbí celkový vzhled aplikace?**

Tuto otázku jsem zvolil, abych zjistil čistě subjektivní názor hodnotícího na celkové grafické zpracování aplikace. Známkování bylo stejné jak u předchozí otázky a i zde bylo průměrné hodnocení velice kladné a to s průměrnou známkou 1.48. Takto kladná známka mne potěšila. Vyplývá z ní, že grafické rozhraní aplikace bylo dobře navrženo i implementováno.

**Otázka: Jak na vás působí barevné schéma aplikace?**

I když je možné nastavit některé barvy aplikace, zajímalo mne hodnocení výchozího barevného schématu. Tato otázka nebyla povinná a bylo na ni možno volně odpovídat. Na otázku odpověděla většina dotázaných. Výchozí barevné schéma bylo hodnoceno kladně až na jednu výjimku. Hodnotící převážně uváděli, že se jim celkové barevné schéma zdá být příjemné a moderní.

### Otázka: Jsou gesta jasná a srozumitelná?

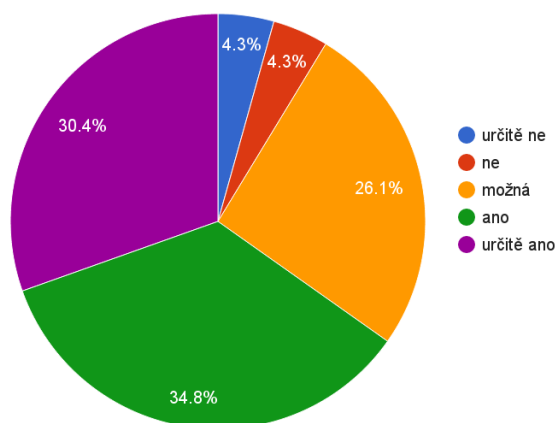
Na tuto otázku bylo možné odpovědět buď možnostmi “ano”, “ne”, nebo vlastní odpovědí. Většina respondentů odpověděla, že popis gest na ně působí srozumitelně. Téměř všichni hodnotící popsaným gestům rozuměli. Pouze 4 respondenti měli připomínky, které se týkaly podobnosti některých gest a nejasnosti popisu gesta pro odsmažení obrázku. Jeden respondent uvedl, že by nejprve musel aplikaci vyzkoušet pro zodpovězení této otázky. Připomínka týkající se gesta odsmažení byla přínosná a budu muset do budoucna vymyslet jakým způsobem toto gesto popsat tak, aby bylo srozumitelnější.

### Otázka: Chybí zde nějaké gesto, které byste osobně využili?

Tato otázka byla volitelná a mohli na ni respondenti odpovídat podle svého uvážení. Mezi návrhy byly například gesto pro uzavření aplikace vyvolané tlesknutím, možnost zobrazení postranních panelů, dále pak zvolení složky s obrázky, nebo možnost prohlížení více obrázků najednou. Jednalo se o velmi užitečné odpovědi a některé nápady budou popřípadě implementovány v budoucím vývoji.

### Otázka: Myslíte si, že ovládání gesty má budoucnost?

V dotazníku jsem chtěl také zjistit pohled hodnotících na tento nový druh ovládání. Jak vyplývá z obrázku 7.3, odpovědi byly tomuto druhu ovládání převážně pozitivně nakloněni.

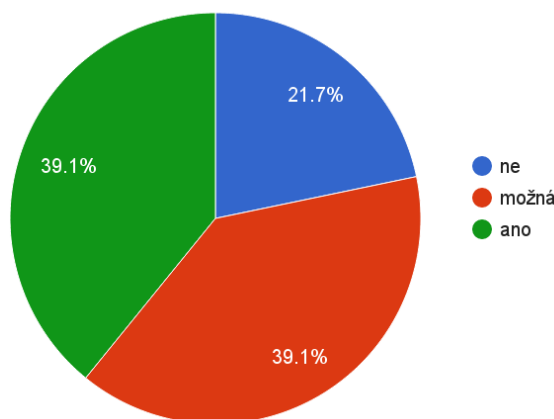


Obrázek 7.1: Graf zobrazující postoj hodnotících k používání bezkontaktního ovládání v budoucnosti.

### Otázka: Využili byste libovolnou aplikaci ovládanou gesty?

Tato otázka byla povinná. Chtěl jsem tímto způsobem zjistit, do jaké míry by respondenti využívali různé aplikace ovládané gesty. Otázka byla však spíše polemická, jelikož se jedná o nový způsob ovládání, na který nejsou uživatelé prozatím běžně zvyklí a tudíž si nejspíš nedovedou uvědomit všechny možnosti tohoto způsobu ovládání. Na obrázku 7.2 jsou zobrazeny odpovědi hodnotících v procentuálním rozložení.





Obrázek 7.2: Graf znázorňující názor na využití i jiných aplikací ovládaných gesty

### Otázka: Máte libovolnou připomínku k aplikaci?

Poslední otázkou jsem chtěl získat názor hodnotících, na libovolnou připomínku k aplikaci. Mezi připomínkami se objevil postřeh, že z tohoto druhu ovládání musí bolet ruka. Což je sice pravda z dlouhodobého užívání, ale při postupném krátkodobém využívání si jde na toto ovládání zvyknout. Další připomínkou bylo, že by aplikace mohla být ovládána plně pomocí gest. O této myšlence jsem již uvažoval a chtěl bych ji implementovat v budoucnu.

## 7.2 Přímé testování

Přímého testování, neboli *Usability Testing* bylo provedeno s 6 uživateli, kteří měli možnost si aplikaci přímo vyzkoušet. Jelikož se aplikace ovládá pomocí jednoduchých úkonů, stále je primárně zaměřena na prohlížení fotek a obrázků. Z tohoto důvodu nelze jednoduše měřit rychlost provedených úkonů, neboť vše závisí na čistě subjektivním vjemu uživatele na právě zobrazeném obrázku nebo fotografii. Toto testování bylo zaměřeno především pro zjištění toho, zda jsou navržená gesta opravdu snadno použitelná. Jednalo se o krátkodobější testování.

Před testováním aplikace byl každému uživateli podrobně popsán způsob práce s aplikací a jednotlivá gesta, která lze využít pro její ovládání. V průběhu testování jsem sledoval chování uživatelů a jejich interakci s aplikací. V případě nejasností jsem jim vysvětlil gesto, kterému nerozuměli. Průběžně jsem si během testování sepisoval důležité poznatky, kterých jsem si všiml.

Po testování jsem se jednotlivých uživatelů zeptal, na celkový dojem z aplikace a jejich pohled na způsob ovládání aplikace gesty. Důležité pro mne byly také připomínky, co by v aplikaci změnili či vylepšili.

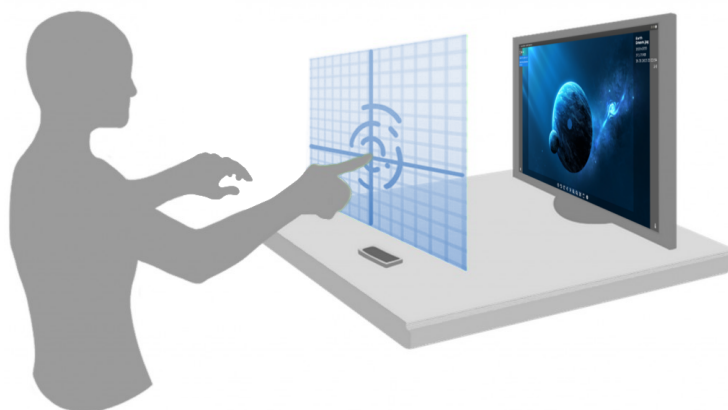
### Zjištěné poznatky

Přímým testováním jsem zjistil mnoho poznatků, které se zde pokusím popsat. Zjistil jsem to, že uživatelům nějakou dobu trvalo než si navykli na celkové ovládání pomocí gest rukou. Toto jsem předpokládal již před testem a tím se mi potvrdil můj předchozí úsudek. Z počátku měli někteří recenzenti přehnanou intenzitu gest a chvíli trvalo než si osvojili

grif určitých gest a to především gest posunu mezi snímky. Po chvíli užívání již ovládali aplikaci o poznání jistěji.

I po krátkém testování se většina testovaných shodla na příliš velké citlivosti některých gest a to především u gesta pro rotaci obrázku. Akce rotace se někdy prováděla namísto zamýšlené operace posunutí. Dále se jim nezdála příliš vhodná podobnost gest pro smazání a přiblížení. Důvodem bylo to, že se během testování občas spínala akce smazání, nebo přiblížení bez přímého úmyslu.

Celkové hodnocení od přímo testujících již nebylo tak pozitivní, jako u testování dotazníkem, což se dalo očekávat. Přineslo však důležité informace a nápady od testujících. Mezi těmito nápady byla kalibrace aplikace před použitím a také možnost nastavení citlivosti gest. Dále také možnost zvolení pouze některých gest jako aktivních, které by uživatel opravdu využil. Nejpodstatnějším zjištěním však především bylo to, že ovládání pouze jednou rukou nemusí být úplně ideální. Celkově toto testování bylo velmi užitečné a přineslo mi nové poznatky o tom, co by bylo vhodné v aplikaci ještě v budoucnu vylepšit.



Obrázek 7.3: Testování aplikace<sup>1</sup>

---

<sup>1</sup>Převzato a upraveno z: <http://www.exeswiss.ch/2013/07/26/test-leapmotion//>

## Kapitola 8

# Závěr

Cílem této práce bylo vytvořit a navrhnout aplikaci prohlížeče fotografií, kterou bude možno ovládat pomocí gest rukou s využitím senzoru *Leap Motion*. Prohlížeč byl navržen tak, aby jej mohl uživatel ovládat pomocí jednoduchých gest rukou. Aplikaci se podařilo úspěšně implementovat a tím splnit zadání práce. Výsledkem práce je tedy funkční multiplatformní aplikace, kterou lze využívat pro prohlížení fotek. Aplikace dále umožňuje uživateli nastavit si některé vlastnosti prohlížeče tak, aby mu co nejvíce vyhovovala. Oproti již existujícím prohlížečům fotografií, které jsou zmíněny v kapitole 3.2 je mnou vytvořená aplikace multiplatformní a lze u ní snadno nastavovat některé vlastnosti. I když aplikace splňuje základní funkcionalitu, chtěl bych i přesto v budoucnu pokračovat v jejím vývoji a rozšiřování celkových možností. Již v průběhu návrhu a implementace mne napadaly způsoby, jak ji vylepšit přidáním nových funkcí. Po testování uživateli tato potřeba vylepšení ještě více vzrostla. Proto zde zmíním některé možné návrhy pro rozšíření a vylepšení stávající aplikace.

V první řadě bych chtěl aplikaci úplně odladit nejen pro platformu *Linux*, ale také na *Windows* a *Mac*. Jak vyplynulo z testování, bude nutná celková úprava některých gest a také vytvoření nových gest především s využitím obou rukou. S gesty souvisí i vytvoření učicího módu, který by novému uživateli poskytl možnost naučit se práci s prohlížečem rychleji. Ovládání pomocí rukou může být po nějaké době unavující, bylo by tedy jistým usnadněním uživateli poskytnout možnost ovládat prohlížeč nástrojem, jako je například tužka nebo ukazovátka. Další vylepšení by se týkalo nastavení, ve kterém by bylo možné nastavit počet zpracovaných *FPS* ze senzoru popřípadě vypnutí animací tak, aby byla aplikace využitelná i na počítačích se slabší technickou výbavou.

Po těchto úpravách jsem také uvažoval o možném uvolnění celé aplikace jako *open source*.

# Literatura

- [1] Global unit sales of video game consoles from 2008 to 2014. [online], 2015-01 [cit. 2015-05-02].  
Dostupné z: <http://www.statista.com/statistics/276768/global-unit-sales-of-video-game-consoles/>
- [2] Cardinal, D.: Kinect for the Xbox One: Sensor revolution or marketing hype? [online], 2013-05-21 [cit. 2015-05-06].  
Dostupné z: <http://www.extremetech.com/gaming/156436-kinect-for-the-xbox-one-sensor-revolution-or-marketing-hype>
- [3] Connie Hu, J. S. . j. t. v.: Magnetometer. [online], [cit. 2015-05-11].  
Dostupné z: <http://arcbotics.com/products/sparki/parts/magnetometer/>
- [4] Gavrylova, N.: Image Viewer Deluxe. [online], 2014-06-26 [cit. 2015-05-12].  
Dostupné z: <https://itunes.apple.com/us/app/image-viewer-deluxe/id662539468?mt=12>
- [5] Husák, M.: Akcelerometry. [online], 2009-01-09 [cit. 2015-05-10].  
Dostupné z: <http://www.micro.feld.cvut.cz/home/x34ses/prednasky/08%20Akcelerometry.pdf>
- [6] Labs, T.: Myo - Tech Specs. [online], 2015 [cit. 2015-05-08].  
Dostupné z: <https://www.thalmic.com/en/myo/techspecs>
- [7] Lamar, M. V.: Hand Gesture Recognition using T-CombNET - A Neural Network Model dedicated to Temporal Information Processing. 2001.
- [8] Leap Motion, I.: PhotoScape - Revolutionary Photo Explorer. [online], [cit. 2015-05-14].  
Dostupné z: <https://apps.leapmotion.com/apps/photoscape>
- [9] Leap Motion, Inc: Leap Motion - Java SDK Documentation. [online], 2015 [cit. 2015-05-13].  
Dostupné z: <https://developer.leapmotion.com/documentation/java/index.html>
- [10] Liang, R.-H.; Ouhyoung, M.: A real-time continuous gesture recognition system for sign language. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, Apr 1998, s. 558–567, doi:10.1109/AFGR.1998.671007.

- [11] Mando: Tutorial - Gyroscope. [online], 2013-02-01 [cit. 2015-05-10].  
Dostupné z: <https://learn.sparkfun.com/tutorials/gyroscope>
- [12] Nikorovič, M.: *Získavanie priestorových obrazov*. bakalárska práca, Univerzita Pavla Jozefa Šafárika v Košiciach, 2012 [cit. 2015-05-09].
- [13] Roebuck, K.: *Gesture Recognition: High-impact Emerging Technology - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors*. Emereo Publishing, 2012, ISBN 9781743045398.  
Dostupné z: <https://books.google.cz/books?id=tJ8NBwAAQBAJ>
- [14] Turner, D.: Hack: The Nintendo Wii. [online], 2007-07-01 [cit. 2015-05-07].  
Dostupné z:  
<http://www.technologyreview.com/hack/408183/hack-the-nintendo-wii/>
- [15] Švrkala, M.: PlayStation Move - revoluce, nebo evoluce? [online], 2010-06-18 [cit. 2015-05-07].  
Dostupné z:  
<http://www.hrej.cz/clanky/playstation-move-revoluce-nebo-evoluce-3156/>

# Příloha A

## Obsah CD

- **README** – Manuál pro spuštění aplikace.
- **projekt.pdf** – Tento dokument ve formátu *PDF*.
- **src/** – Veškeré zdrojové soubory projektu.
- **app/** – Spustitelný soubor *jar* s potřebnými knihovnami.
- **doc/** – Dokumentace projektu vytvořenou nástrojem *JavaDoc*.
- **video/** – Video prezentující funkční aplikaci.
- **tex/** – Zdrojové soubory tohoto dokumentu v  $\text{\LaTeX}$ .
- **form/** – Dotazník a získaná data z něj.