

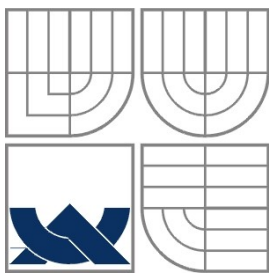
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

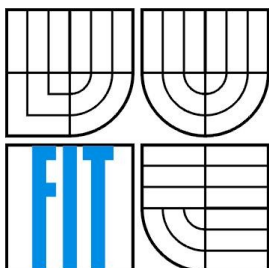
BAKALÁŘSKÁ PRÁCE

Brno, 2016

Michael Švasta



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

UŽIVATELSKÉ ROZHRAŇÍ PRO OVLÁDÁNÍ 3D APLIKACÍ ALE I 2D APLIKACÍ VE 3D

USER INTERFACE FOR 3D APPLICATIONS
AND ALSO 2D APPLICATION IN 3D

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAEL ŠVASTA

VEDOUCÍ PRÁCE

SUPERVISOR

Prof. Dr. Ing. PAVEL ZEMČÍK

BRNO 2016

Abstrakt

Cílem této bakalářské práce je navrhnout a implementovat program, který je možné ovládat za použití bezdotykového ovládání. Práce obsahuje popis technologie Leap motion a možnosti vývoje pro toto zařízení. Dále je představeno uživatelské rozhraní aplikace a následně jsou detailně popsány jeho jednotlivé části. V poslední sekci práce jsou prezentovány metody testování a zhodnocení výsledků testů.

Abstract

The goal of this bachelor's thesis is to design and implement a software, which can be controlled by a using hands free gestures. This thesis contains a description of the Leap motion technology and possibilities of development for this device. Further, there is introduction of graphical user interface and then there is detailed description of its individual parts. Last section of thesis presented methods of testing and results of these tests.

Klíčová slova

Leap motion, bezdotykové ovládání, ovládání gesty, kreslení, kreslicí editor, grafické uživatelské rozhraní

Keywords

Leap motion, hands free controller, gesture controller, painting, painting editor, graphical user interface

Citace

Michael Švasta: Uživatelské rozhraní pro ovládání 3D aplikací (ale i 2D aplikací ve 3D), bakalářská práce, Brno, FIT VUT v Brně, 2016

Uživatelské rozhraní pro ovládání 3D aplikací

Ale i 2D aplikací ve 3D

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Prof. Dr. Ing. Pavla Zemčíka. Další informace k práci jsem čerpal z literárních pramenů a publikací uvedených v seznamu literatury.

.....
Michael Švasta
15. 5. 2016

Poděkování

Rád bych touto cestou poděkoval mému vedoucímu práce Prof. Dr. Ing. Pavlu Zemčíkovi, který mě vedl k cíli a inspiroval mě při realizaci. Dále jsem vděčný všem, kteří se ochotně zapojili do testování aplikace a podělili se se mnou o své postřehy a námítky.

© Michael Švasta, 2016.

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

1 Úvod.....	2
2 Stávající situace bezdotykového ovládání.....	3
2.1 Bezdotykové ovládání.....	3
2.2 Architektura Leap motion.....	5
2.3 Vývoj pro Leap motion.....	8
2.4 Orion.....	15
2.5 Existující aplikace.....	17
3 Zhodnocení současného stavu.....	20
3.1 Zhodnocení technologií.....	20
3.2 Téma aplikace.....	21
3.3 Popis a cíle aplikace.....	21
3.4 Zhodnocení programu.....	22
3.5 Návrh budoucí práce.....	22
4 Implementace.....	23
4.1 Vývoj programu.....	23
4.2 Kreslicí editor.....	23
4.3 Spojení s Leap motion.....	25
4.4 Ovládání aplikace.....	26
4.5 Nástroje.....	28
4.6 Panel barev.....	36
4.7 Informační lišta.....	38
4.8 Hlavní nabídka.....	39
5 Testování.....	40
5.1 Alfa verze.....	40
5.2 Beta verze.....	41
5.3 Zhodnocení testování.....	41
6 Závěr.....	43
Literatura.....	44
Seznam obrázků.....	46
Seznam algoritmů.....	48
A.1 Dotazník.....	49
A.2 Vyhodnocení dotazníku.....	50
B Obsah CD.....	52

1 Úvod

Se stále větším zapojováním výpočetní techniky do běžného života je nutné nalézat lepší a efektivnější metody, jak ovládat počítač, aniž by bylo nutné využívat klasické periferie, kterými jsou klávesnice a myš. Možnosti bezdotykového ovládání jsou omezovány schopnostmi hardwaru, ale i tyto limity jsou rok od roku posouvány.

Vhodným použitím bezdotykového ovládání můžeme usnadnit život pohybově omezeným lidem, např. použitím hlasového nebo očního ovládání, dále v provozech, kde zaměstnanci pracují s enormně mastnými či špinavými rukama. Bezdotykové ovládání je rovněž možné použít při interaktivní výuce či v neposlední řadě jen pro zpestření zábavy.

Téma jsem si zvolil, jelikož jsi myslím, že vhodným použitím bezdotykového ovládání lze dosáhnout lepší efektivity a může působit přirozeněji než při využití klávesnice a myši.

Cílem práce je vytvořit program, který by byl ovládán zařízením Leap motion tak, aby ergonomie používání a celkově ovládání programu byla vyšší než s myši. To zahrnuje vytvoření vhodných gest, která by byla jednoduchá, ale účelná, a uživatel by se snadno naučil princip ovládání.

Důvodem, proč jsem si zvolil toto zadání, je, že mě zajímají nové technologie a také jsem se chtěl více seznámit s problematikou bezdotykového ovládání, jelikož se domnívám, že je to určitá cesta do budoucna.

Práce je rozdělena do šesti kapitol. V následující kapitole je text věnován technologiím bezdotykového ovládání, možnostmi vývoje pro Leap motion a existujícím aplikacím pro zařízení. Ve třetí kapitole jsou jednotlivé technologie porovnány a rovněž je zde uveden návrh aplikace. Čtvrtá kapitola popisuje použité vývojové prostředí včetně programovacího jazyka a celkovou implementaci aplikace. V předposlední kapitole jsou přiblíženy jednotlivé fáze testování a jejich zhodnocení. Poslední kapitola se zamýšlí nad samotnou prací a jejími výsledky, dále využitelností bezdotykových technologií a také možným budoucím rozšíření práce.

2 Stávající situace bezdotykového ovládání

Tato kapitola je zaměřena na poskytnutí informací o jednotlivých technologiích souvisejících s touto prací. Poznatky jsou čerpány z literatury, obrázky jsou převzaté z internetu, pokud není uvedeno jinak.

Kapitola je rozdělena do několika částí. Nejprve jsou popsány vybrané technologie, jež umožňují bezdotykové ovládání počítače a jsou na trhu k dostání i koncovým uživatelům. Zbylé části kapitoly se zabírají konkrétní technologií, a to Leap motionem, kde jsou popisovány základní vlastnosti a možnosti vývoje pro zařízení, dále další možnosti využití a v závěru kapitoly jsou popsány již existující aplikace.

2.1 Bezdotykové ovládání

Existují různé technologie pro usnadnění práce s počítačem, které se na trhu začaly v posledních letech objevovat. Jednotlivá zařízení se od sebe liší a pracují na jiném principu.

- **Leap motion**

Leap motion[1] je technologie pro ovládání počítače pomocí rukou a prstů. Dvě monochromatické kamery sledují prostor nad zařízením, jenž je osvětlován třemi infračervenými LED. Získaná data jsou přes USB přenesena do počítače a výsledná scéna, respektive model rukou, je sestavena softwarem v počítači s využitím složitých matematických výpočtů. Vše je umístěno v malé krabičce o velikosti několika centimetrů. [2]



Obrázek 1: Leap motion¹

Tvůrci se zaměřili na velmi malou oblast snímání, avšak na druhou stranu jim to umožnilo dosáhnout velmi rychlou odezvu s až 300 snímky za sekundu.

Zařízení se dostalo na trh v červenci 2013, třebaže technologie byla vyvinuta již v roce 2008.

1 Zdroj: PC Mag: <http://www.pcmag.com/article2/0,2817,2422045,00.asp>

- **Kinect**

Kinect je technologie vyvíjená firmou Microsoft, která měla nabídnout majitelům konzolí Xbox novou dimenzi zábavy. Chyba: zdroj odkazu nenalezen



Obrázek 2: Kinect²

Princip je založen na infračervené projekci, která je snímána širokoúhlou kamerou s rozlišením 1080p. Avšak Kinect umožňuje sledovat celé tělo a přidružené okolí, a tudíž musí být schopen zpracovat 2 gigabity dat za sekundu. Zařízení dokáže sledovat až 6 lidí najednou, tepovou frekvenci a gesta vytvářená ovladačem Xbox One.

Nedílnou součástí zařízení je rovněž mikrofon, kterým mohou být přijímány hlasové povely, i když je konzole v režimu spánku.[3]

- **Tobii EyeX**

Tobii EyeX, vycházející z Tobii gaze, jež bylo poprvé uvedeno na CESu 2013, je vyvíjeno švédskou společností Tobii[5], jež se specializuje na technologie sledování očí.



Obrázek 3: Tobii EyeX³

2 Zdroj: Microsoft: <http://www.xbox.com/en-US/xbox-one/accessories/kinect-for-xbox-one>

3 Zdroj: Expert Reviews: <http://www.expertreviews.co.uk/accessories/gadgets/1403236/steels-series-sentry-review-eyes-on-with-tobiis-eye-x-tech-and-assassins>

Jedná se o zařízení, které je umístěno na spodní straně monitoru, a umožňuje ovládat kurzor myši pouhým pohledem. Rozpoznávání funguje bez ohledu na okolní osvětlení, jelikož snímač má vlastní zdroj paprsků, jež se odrážejí v očích.[6]

- **MYO**

MYO je náramek, vytvořený firmou Thalmic Labs, umístěný na horní části předloktí, kterým lze díky integrovaným sensorům velmi rychle a snadno sledovat nejen úhly natočení ruky ve všech směrech, ale také různé kombinace stisku prstů, jelikož se to projeví jinou formou stlačených svalů, na kterých je náramek umístěn.[7]



Obrázek 4: MYO⁴

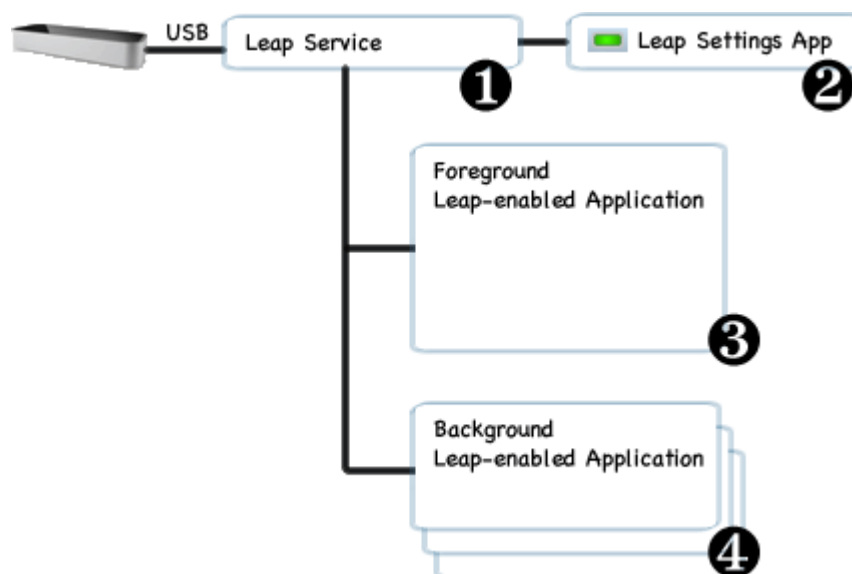
2.2 Architektura Leap motion

Software Leap motionu běží jako služba na systémech MS Windows nebo jako démon na systémech, které vycházejí z Unixu. Software se se zařízením spojí skrze USB sběrnici. Aplikace, využívající Leap motion, poté přistupují k službě nebo démonu, které přijímají data ze kontroléru. Leap motion SDK nabízí dvě varianty aplikačního rozhraní (API) pro sběr dat – přirozené rozhraní (*native interface*) a rozhraní WebSocket (*WebSocket interface*). Tato API umožňují vytvoření aplikací v několika programovacích jazycích.[8]

4 Zdroj: The Wall Street Journal: <http://blogs.wsj.com/personal-technology/2014/06/10/myo-motion-sensing-aramband-first-look-at-the-wearables-finished-design/>

Nativní rozhraní

Toto rozhraní je poskytováno skrze dynamicky načítané knihovny. Tato knihovna se připojí ke službě (démonu) Leap motionu a poté poskytuje snímaná data aplikaci. Knihovna lze napojit přímo v C++ a Objective-C aplikacích, nebo pomocí jedné z jazykových vazeb pro Javu, C# a Python.



Obrázek 5: Native interface⁵

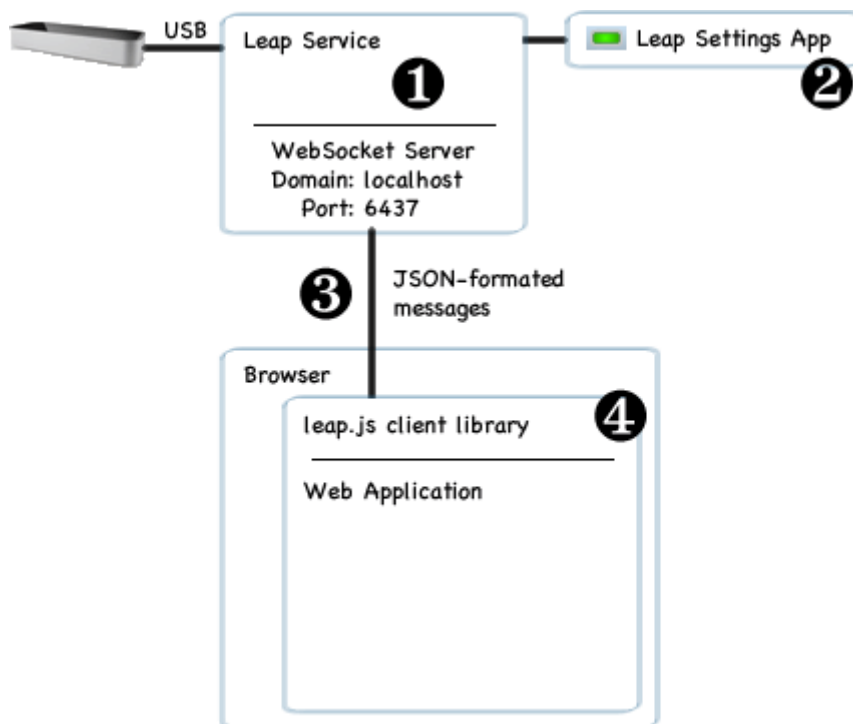
Popis:

1. Služba přijímá data z kontroléru přes USB, která zpracuje a pošle je běžící aplikaci využívající Leap motion (*Leap-enabled Application*). Ve výchozím stavu služba posílá informace pouze do aplikace v popředí, avšak aplikace může požádat, aby data byla přijímaná na pozadí.
2. Konfigurační aplikace Leap motinu běží odděleně od služby a umožňuje uživateli konfigurovat zařízení.
3. Aplikace využívající Leap motion na popředí přijímá snímaná data ze služby. Aplikace se ke službě může připojit pomocí knihovny.
4. Pokud se aplikace dostane na pozadí, služba přestane vysílat data. Jestliže aplikace vyžaduje data i na pozadí, může si požádat o povolení k přijímání dat, i když okno nebude aktivní.

⁵ Zdroj: Leap Motion: https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Architecture.html

Rozhraní WebSocket

Leap motion služba běží jako WebSocket server na doméně localhost (127.0.0.1) s portem 6437. Rozhraní poskytuje snímaná data ve formě JSON zpráv. Knihovna JavaScriptového klienta zpracuje zprávy a snímaná data prezentuje jako regulární JavaScriptové objekty.



Obrázek 6: WebSocket interface⁶

Popis:

1. Služba poskytuje server na <http://127.0.0.1:6437>.
2. Konfigurační aplikace umožňuje koncovému uživateli spuštění a zastavení WebSocket serveru.
3. Server posílá snímaná data ve formě JSON zpráv, aplikace může zaslat zpět serveru konfigurační zprávu.
4. Klientská JavaScriptová knihovna *leap.js* by měla být použita ve webové aplikaci. Knihovna vytváří spojení k serveru a zpracovává JSON zprávy.

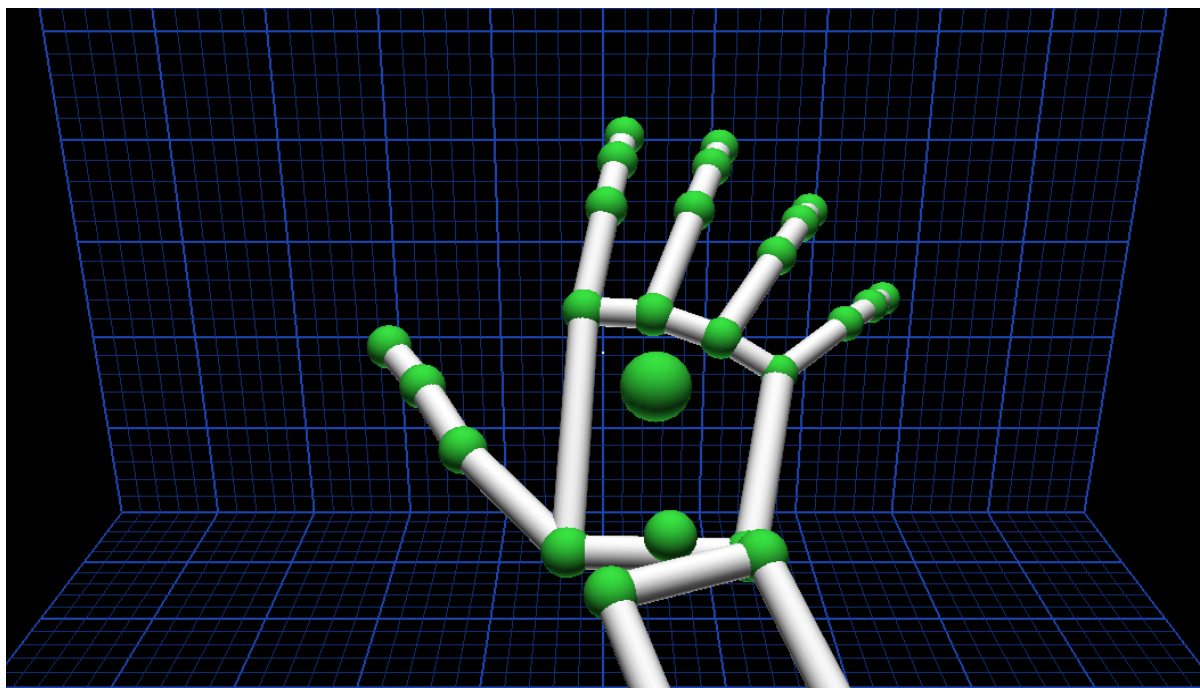
⁶ Zdroj: Leap Motion: https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Architecture.html

2.3 Vývoj pro Leap motion

Pro vývoj je k dispozici SDK, tedy sada vývojových nástrojů, které obsahuje potřebné knihovny pro ovládání zařízení. Jelikož se firma snaží oslovit co nejvíce vývojářů, je SDK dostupné pro různé programovací jazyky a dokumentace k SDK[9][10] je rozsáhlá, ale přehledná, a poskytuje všechny důležité informace.

Visualizer

Součástí ovladačů je aplikace Visualizer, jež přehledně zobrazuje právě zachycená data na zařízení a je vhodným nástrojem pro kalibraci zařízení či ladění programu.



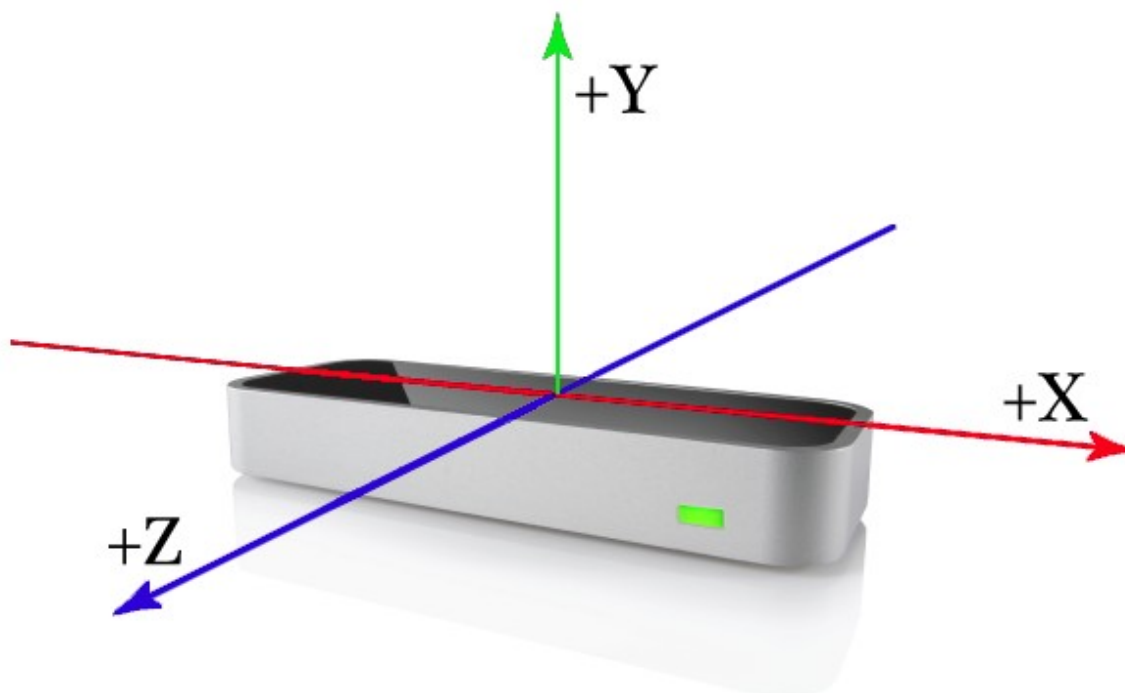
Obrázek 7: Podoba Visualizeru⁷

Souřadnicový systém

Systém Leap motion využívá pravotočivou soustavu prostorových kartézských souřadnic. Počátek se nachází ve středu horní části zařízení. Osy X a Z jsou horizontální s tím, že osa x je rovnoběžná s delší stranou zařízení, zatímco osa z je kolmá na delší stranu a její kladné hodnoty jsou ve směru k uživateli. Vertikální osou je Y a její kladná část se nachází nad snímačem.

Leap motion API využívá metrické jednotky, přesněji milimetry pro vzdálenost, mikrosekundy pro čas, milimetry za sekundu pro rychlost a radiány pro úhel.

⁷ Zdroj: Leap Motion: Program Visualizer



Obrázek 8: Pravotočivý souřadnicový systém⁸

Obrazovka

Kontrolér má jednu nebo více kalibrovaných obrazovek (*calibratedScreens*), což je počítačem zobrazené zorné pole kontroléru, který zná pozici a dimenzi. Díky toho je možné určit, jakým směrem uživatel ukazuje.[11]

Snímaná data

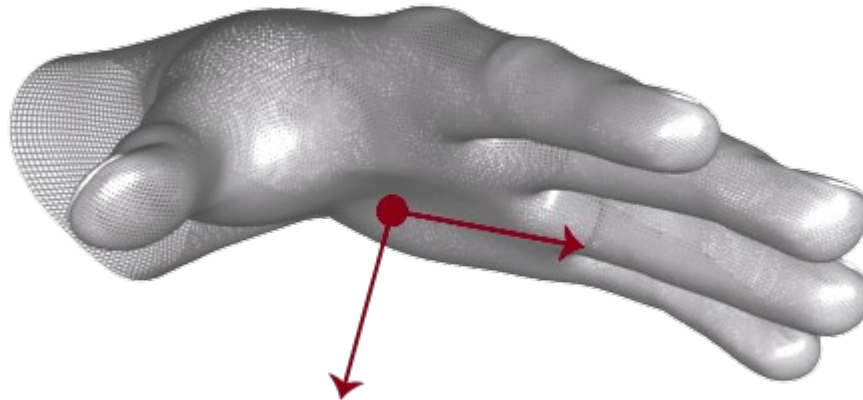
Kontrolér sledující ruce a prsty v jeho zorném poli, neustále poskytuje aktualizované snímky dat. Každý snímek (*Frame*) je objekt, který obsahuje seznam zachycených entit, jako jsou ruce, prsty a nástroje, rovněž rozpoznaná gesta, či faktory, jež celkově popisují scénu. Objekt *Frame* je základem datového modelu Leap motion.

Ruce

Model ruky poskytuje informace o identitě, pozici a jiných vlastnostech, případně, o jakou ruku se jedná a seznam prstů.

Ruce jsou reprezentovány třídou *Hand*.

⁸ Zdroj: Leap Motion: https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Overview.html



Obrázek 9: Orientace ruky je definována normálovým vektorem dlaně a jejím směrem⁹

Software Leap motionu využívá vnitřní model lidské ruky pro poskytnutí pravděpodobného umístění, i když část ruky není viditelná. Model vždy zajistí pozici všech pěti prstů, i když se ruka nemusí nacházet v optimální poloze, aby byly všechny prsty viditelné. Software využívá viditelné části, interní model a poslední pozorování pro výpočet nejpravděpodobnější pozice všech částí, přestože nemusí být viditelné.

Paže

Paže je objekt *Arm*, který zajišťuje orientaci, délku, šířku a koncový bod paže. Pokud není loket v záběru, jeho pozice se odhadne pomocí posledních sledování a z typických proporcí.

Prsty

Kontrolér poskytuje informace o každém prstu na ruce. Pokud jakákoliv část prstu není viditelná, je konečná podoba založena na nedávných snímcích a anatomickém modelu ruky. Prsty jsou identifikovány anglickými jmény, tedy *thumb*, *index*, *middle*, *ring*, and *pinky*.

Prsty jsou reprezentovány třídou *Finger*, která je součástí objektu *Pointable*.

9 Zdroj: Leap Motion: https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Overview.html



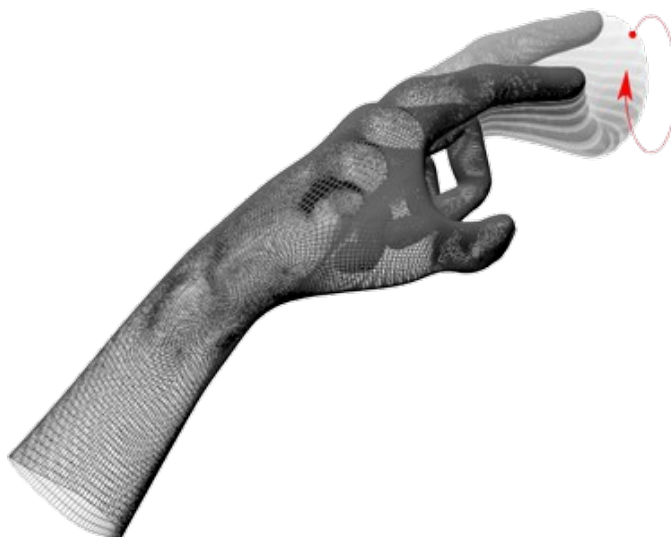
Obrázek 10: Pozice a směr je určen konečky prstů¹⁰

Gesta

Zařízení rozpoznává určitý pohybový vzor jako gesto, které může být indikováno jako příkaz. Gesta jsou posuzována pro každý prst individuálně. V API se nachází čtyři základní gesta pro ovládání a to kruhové gesto (circle), švihnutí (swipe), klepnutí na klávesu (key tap) a klepnutí na obrazovku (screen tap). Pro gesta jsou rovněž rozpoznávané základní vlastnosti, jako o kterou ruku se jedná, který prst aj.

- **Gesto Circle**

Gesto reprezentuje krouživý pohyb prstu a je realizováno třídou *CircleGesture*. Gesto je rozpoznáno v momentě, kdy koneček prstu kreslí kruh uvnitř zorného pole zařízení.



Obrázek 11: Circle - prst krouží po kružnici¹¹

10 Zdroj: Leap Motion: https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Overview.html

11 Zdroj: Leap Motion: https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Overview.html

Gesto musí být nepřetržité a může nabývat tří stavů:

STATE_START – gesto začalo a pokračovalo dostatečně dlouho pro rozpoznání kruhu

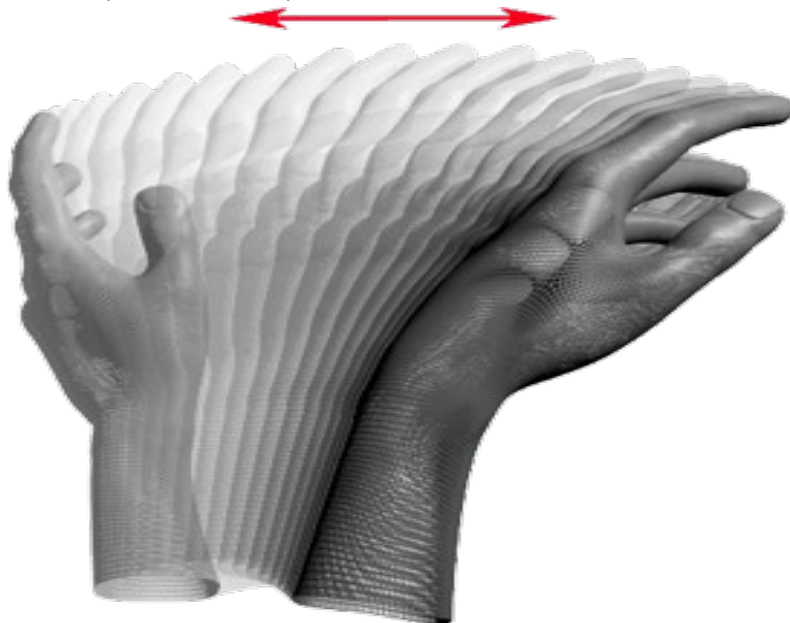
STATE_UPDATE – kruhové gesto pokračovalo

STATE_STOP – gesto bylo ukončeno

Pro gesto lze nastavit parametry minimální radius a minimální opsaný úhel.

- **Gesto Swipe**

Třída *SwipeGesture* implementuje gesto pro švihnutí prstu. Gesto je generováno pro každý viditelný prst a podle identifikátoru je aktualizováno v každém snímku. Pro švihnutí mohou být nastaveny parametry minimální délky a minimální rychlosti.¹²



Obrázek 12: *Swipe* - dlouhý lineární pohyb ruky a prstů¹²

- **Gesto Key tap**

Třída realizující gesto je *KeyTapGesture*. Gesto simuluje stisknutí klávesy a před jeho provedením je nutné, aby se prst na chvíli ustálil v jedné poloze. Gesto může být ovlivněno parametry minimální rychlostí směrem dolů, minimální vzdáleností a dobou pohybu.

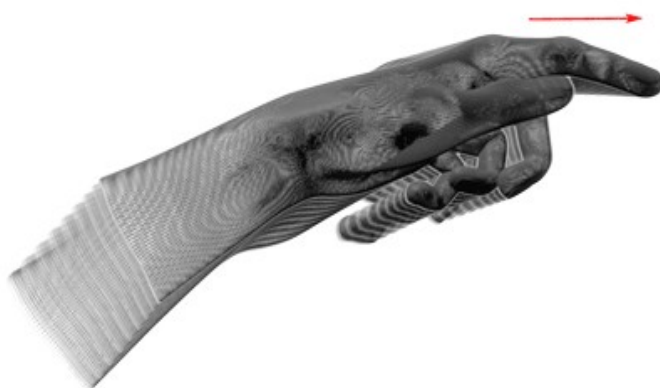
12 Zdroj: Leap Motion: https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Overview.html



Obrázek 13: Key tap - simulace stisku tlačítka¹³

- **Gesto Screen tap**

Gesto simuluje stisknutí obrazovky a jeho implementace se nachází v třídě *ScreenTapGesture*. Před samotným pohybem je nutné ustálit prst v jedné pozici. Gesto může být ovlivněno parametry minimální rychlostí směrem dolů, minimální vzdáleností a dobou pohybu.¹³¹⁴



Obrázek 14: Screen tap - simulace klepnutí na obrazovku¹⁴

Pointable

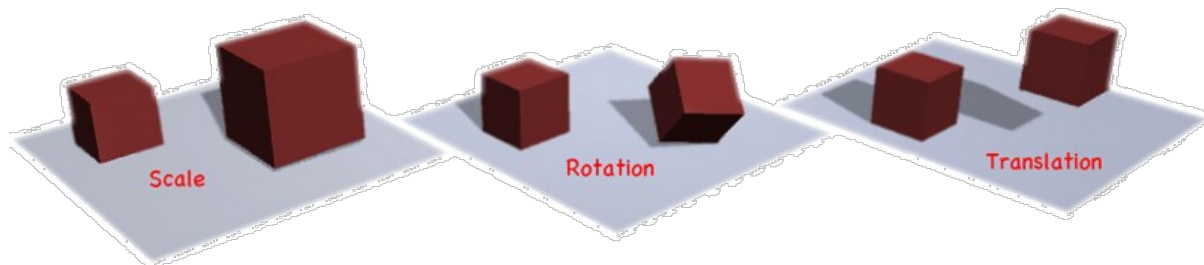
Třída *Pointable* reprezentuje objekty, jež jsou schopné na něco ukazovat, a tím není myšleno jenom prsty, ale i nástroje. Ke klasifikaci detekované entity jako nástroje dojde, pokud je užší, delší a přímější než typický prst. Nejdůležitějšími atributy zde jsou pozice konce ukazovátka a směr.

13 Zdroj: Leap Motion: https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Overview.html

14 Zdroj: Leap Motion: https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Overview.html

Pohyby

Pohyby jsou odhadnuty ze základních typů pohybových vlastností ze změny uživatelských rukou v čase. Pohyby zahrnují měřítko (*scale*), rotaci (*rotation*) a změnu polohy (*translation*).

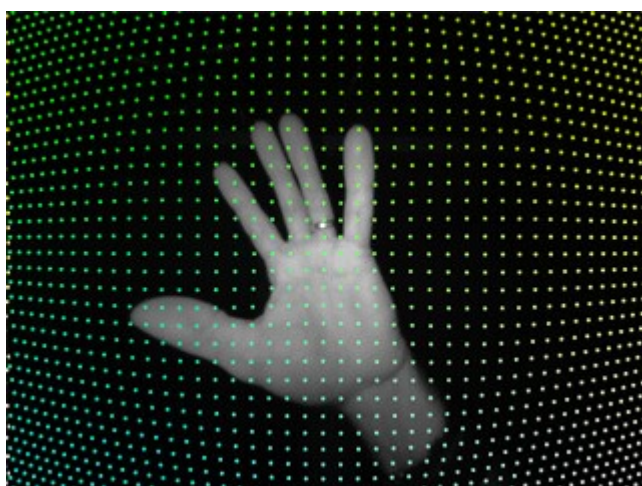


Obrázek 15: Vlastnosti pohybu¹⁵

Pohyby jsou počítány mezi dvěma snímky. K těmto faktorům scény je možné přistupovat jako k celku z objektu snímku, nebo k jednotlivým vlastnostem, které jsou spojeny s objektem ruky. Tyto faktory mohou být sloužit k interakci uvnitř aplikace, kdy, místo sledování směny pozice jednotlivých prstů skrze několik snímků, lze použít měřítkový faktor, který je spočítán mezi dvěma snímky, ke zjištění změny velikosti objektu.

Snímek senzoru

Spolu s počítanými daty lze získat i snímek přímo z kamery v zařízení. Data obrázku obsahují změřenou hodnotu infračerveného jasů a kalibrační data vyžadována pro opravu komplexního zkreslení objektivu. Snímek senzoru lze využít pro aplikace rozšířené reality, zvláště pokud je Leap motion kontrolér připevněn rovnou k headsetu VR.



Obrázek 16: Snímek senzoru¹⁶

15 Zdroj: Leap Motion: https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Overview.html

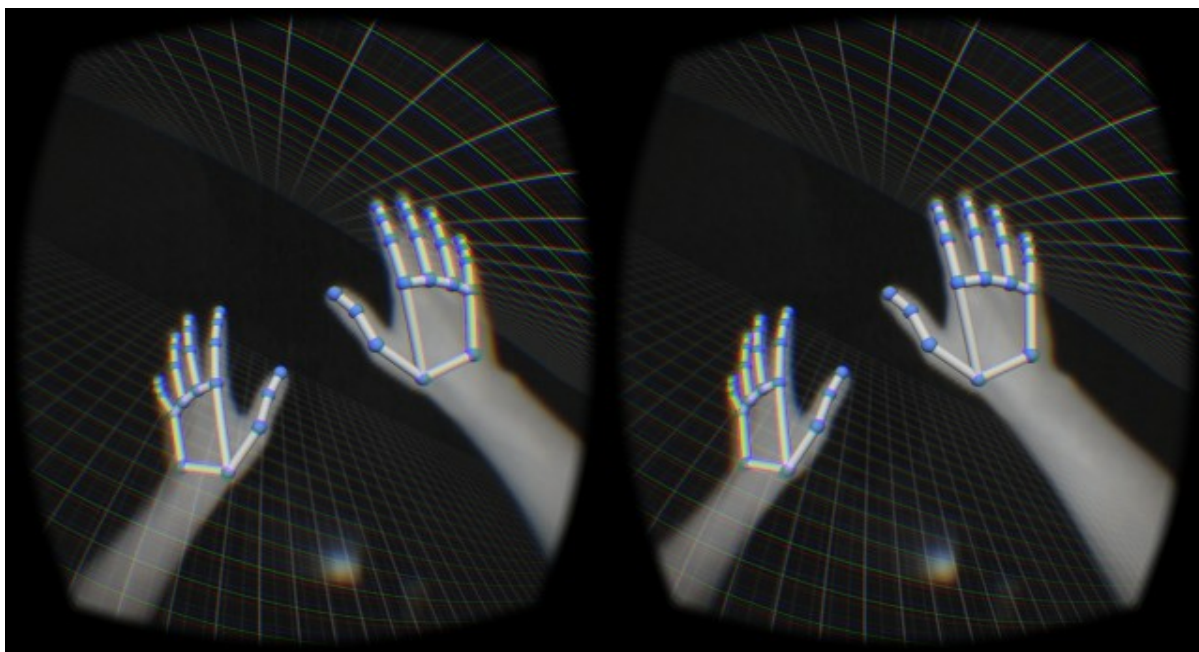
16 Zdroj: Leap Motion: https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Overview.html

2.4 Orion

Po prvním uvedení zařízení v roce 2012 se vytvořila vlna zájmu, která brzy opadla, jakmile lidé zjistili, že v současné podobě použitelnost technologie pokulhává. Kvalitních aplikací, které by dokázaly využít potenciál zařízení, bylo poskrovnu a myšlenka „dělej věci na počítači, jako je děláš v reálném životě“ se neujala, jelikož lidé již objevili ideální způsob interakce – klávesnice a myš.

Avšak poté se na trhu objevila nová zařízení pracující s virtuální realitou, pro která byl klasický způsob ovládání nevhodný, a rázem Leap motion a podobná zařízení dostala smysl.

Z toho důvodu se společnost začala více věnovat virtuální realitě a tím dala vzniknout projektu Orion[13], který oficiálně představila v únoru 2016. Jedná se o sledovací systém ušitý na míru zařízením pro virtuální realitu a malý senzor může být vestavěn přímo do nich. Zatímco hardware je určen pouze pro výrobce, beta verze softwaru je dostupná pro zařízení Leap motion. Orion software přináší nová vylepšení ve sledování rukou, která mají za následek nižší odezvu a lepší a rychlejší rozpoznávání horních končetin.[15]



Obrázek 17: Leap motion ve spojení s Oculus riftem¹⁷

Společnost rovněž na trh uvolnila nástavec, pomocí kterého je možno připevnit Leap motion k Oculus Rift. Přichycení pomáhá zafixovat Leap motion kontrolér v pohledu ven z headsetu, a tím umožňuje promítat infračervené pole před uživatele zařízení. Tento způsob optimalizuje funkci senzorů, jelikož dochází ke sledování v módu shora-dolů, namísto klasického desktopového využití zdola-nahoru.

17 Zdroj: Ars Technica: <http://arstechnica.com/gadgets/2014/08/leap-motion-gets-a-head-mounted-upgrade-becomes-a-vr-controller/>



Obrázek 18: Přichycení Leap motion k Oculus riftu¹⁸

Zorné pole Leap motinu je širší než u Oculusu, což umožňuje mapování uživatelových rukou kdekoli vnitř vizuálního prostoru Oculusu. Tím je dáno realističtější pojetí, jelikož když jsou ruce viditelné, mohou být i použité.[14]



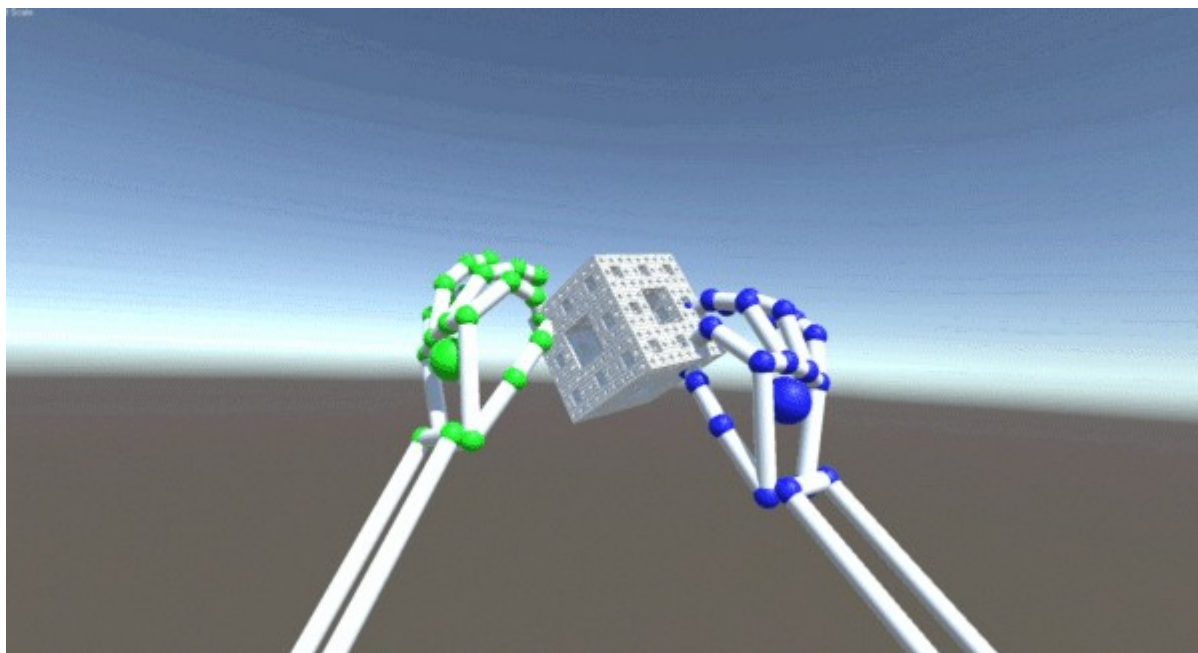
Obrázek 19: Zorná pole Leap motinu a Oculus riftu¹⁹

18 Zdroj: Leap Motion: <https://www.leapmotion.com/product/vr>

19 Zdroj: Ars Technica: <http://arstechnica.com/gadgets/2014/08/leap-motion-gets-a-head-mounted-upgrade-becomes-a-vr-controller/>

Pinch Utility

Pinch utilities je nový modul pro software Orion, který dává uživateli přístup k možnostem sevření a chytání. Utilita je základní komponentou pro detekování pinch gesta a poskytuje o něm užitečné informace. S radikálním rozšířením snímacích schopností Orionu přináší tato funkčnost možnosti interakce s digitálním obsahem jako nikdy předtím.[16]



Obrázek 20: Pinch Utilities²⁰

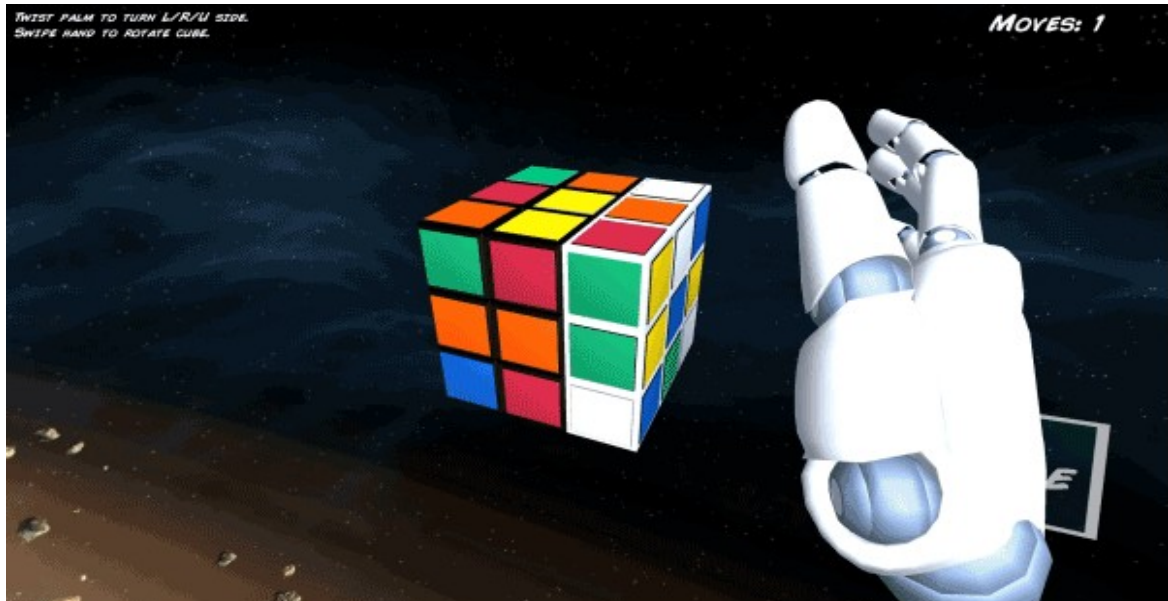
2.5 Existující aplikace

V této části textu jsou popsány vybrané aplikace, které mě určitým způsobem zaujaly, a lze na nich vidět možnosti použití bezdotykového ovládání.

Cube simulator

Jedná se o klasickou Rubikovu kostku, která je však ovládána s využitím Leap motionu. Ovládání je v celku jednoduché, stačí umístit dlaň nad levou, pravou nebo horní stranu kostky pro otočení zvolené řady. V případě, že nutné kostku otočit, lze sklouznout rukou zleva doprava a naopak pro otočení kostky v daném směru.[17]

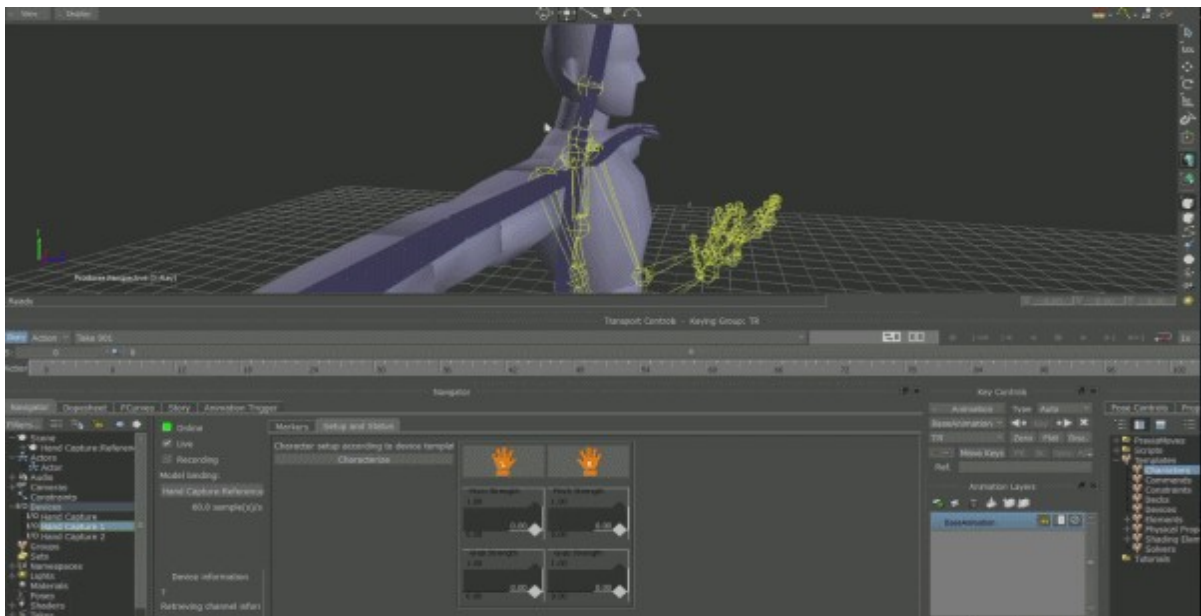
²⁰ Zdroj: Leap Motion: <http://blog.leapmotion.com/power-fingertips-pinch-utilities-orion/>



Obrázek 21: Cube simulator²¹

Hand Capture

Hand Capture je plugin pro Autodesk MotionBuilder 2016. Plugin využívá kontrolér Leap motion jako vstupní periférii pro zachycení pohybu rukou a prstů v reálném čase uvnitř Autodesk MotionBuilder, což je specializovaný nástroj pro zpracování animací a virtuální produkci.[18]



Obrázek 22: Hand Capture²²

21 Zdroj: Leap Motion: <https://developer.leapmotion.com/gallery/cube-simulator>

22 Zdroj: Leap Motion: <https://developer.leapmotion.com/gallery/hand-capture>

Zachycený pohyb může být spojen s rukama a prsty modelované 3D postavy, nebo jej lze propojit s rozdílnými vlastnostmi jakéhokoliv jiného objektu, například s pohybem očí a pusy postavy, nebo barvou a průhledností materiálu objektu. Plugin rovněž dokáže zpracovat sílu stisku, kterou je možné přenést do MotionBuilderu.

Pinch Draw

Tato aplikace pro virtuální realitu pracuje s Oculus Riftem a využívá software Orion, který je popsán v 2.4. Pinch Draw je postaveno na Pinch Utilities, více v 2.4. V tomto jednoduchém demu lze vytvořit náčrtky ve vzduchu, simulováním kreslení s prsty držící štětec. Jednoduchá interakce, jako je tato, může být mocná pro vytváření více komplexních nástrojů. [19]



Obrázek 23: Pinch Draw²³

23 Zdroj: Leap Motion: <https://developer.leapmotion.com/gallery/pinch-draw>

3 Zhodnocení současného stavu

V této části jsou porovnány technologie z předcházející kapitoly se subjektivním ohodnocením jejich vhodnosti pro daný účel. Dále je přiblížen stávající průběh práce a v závěru je uveden návrh, jak by šla práce ještě vylepšit.

3.1 Zhodnocení technologií

Pro tuto práci je důležitý výběr vhodné technologie pro bezdotykové ovládání. Z dostupných zařízení má každé své výhody a nevýhody. Pro výběr bylo důležité, zda je zvolená technologie dostupná českým uživatelům a za jakou cenu.

	Leap motion	Kinect	Tobii EyeX	MYO
Snímaná oblast	Prsty, ruce	Celé tělo	Oči	Celá ruka
Dostupnost v ČR	Ano	Ano	Ne	Ne
Cena²⁴	1301,17 Kč	2365,96 Kč	3215,62 Kč	4708,74 Kč
Vhodnost	Ano	Spíše ano	Spíše ano	Spíše ano

- **Leap motion**

Jak již bylo zmíněno v předchozí kapitole, z vybraných technologií bylo zvoleno Leap motion, které při porovnání ceny a možnosti využití se subjektivně jevílo nejlépe. Dalším důvodem byla rovněž dostupnost zařízení, jelikož společně s Kinectem je k dostání i v českých obchodech.

- **Kinect**

Kinect nabízí největší snímanou oblast, což se projevuje na přesnosti v porovnání s ostatními technologiemi. Přestože je na českém trhu z daných zařízení nejlépe dostupné, typ snímání není vhodný pro danou aplikaci.

- **Tobii EyeX**

Tobii EyeX se může jevit jako nejlepší technologie pro hendikepované lidi, avšak přesnost ovládání hodně závisí na vlastnostech oka, jako je velikost či barva. Rovněž v současné době není zařízení přímo dostupné v České republice.

²⁴ Cena je převzata z oficiálních stránek výrobce s kurzem 23,662 USD a 27,022 EUR

- **MYO**

MYO je svou přesností vhodné k prakticky jakémukoliv použití při ovládní počítače díky přesnému monitorování šlach ruky, avšak pro daný účel není vhodné z důvodu vysoké ceny a nedostupnosti na českém trhu.

3.2 Téma aplikace

Pro využití technologie Leap motion jsem zvolil vytvoření programu kreslicí editor, jelikož se domnívám, že právě kreslení je vhodné pro bezdotykové ovládní, a rovněž jsem chtěl naprogramovat vlastní editor, který by dosahoval lepších kvalit než výchozí nástroj dostupný ve Windows. Kreslení ve 2D by se dalo realizovat v osách X a Y, čímž by se řídil pohyb kurzoru, a třetí osa by mohla ovládat šířku kreslené čáry.

3.3 Popis a cíle aplikace

Hlavním cílem je vytvořit program, jenž by umožňoval snadné kreslení či úpravu grafiky. To zahrnuje navrhnutí přehledného prostředí s důrazem na intuitivní a možná co nejjednodušší ovládní.

Návrh okna editoru, jenž by se skládalo ze tří hlavních částí:

- kreslicí plátno
- panel nástrojů
- informační lišta



Obrázek 24: Návrh okna aplikace

3.4 Zhodnocení programu

Ve stávajícím průběhu implementace se podařilo vytvořit kompletní kreslicí editor, který je schopen kreslit nejen čáru, ale i různé tvary v různých barvách. Pohyb kurzoru je ovládán pravým ukazováčkem. Zahájení kreslení se provede spojením ukazováčku s prostředníčkem a pozastavení kreslení opětovným rozpojením těchto prstů.

Pro výběr nástroje z nástrojové lišty slouží gesto švihnutí do strany, které je v momentální realizaci příliš citlivé a skáče přes více nástrojů.

Stejný problém nastal i u výběru jedné ze základních šesti barev, kde je rovněž využíváno gesto švihnutí, ale ve směru nahoru a dolů.

Aktuálně je největší problém v citlivosti zařízení, které přenáší na kreslicí plátno i chvějící se prst, z čehož vznikne kostrbatá čára.

3.5 Návrh budoucí práce

V návrh budoucí práce bude důležité implementovat funkce, které by nejen usnadňovaly práci s editorem, ale rovněž by poskytovaly přesnější, respektive méně citlivé, ovládání.

- Nejdůležitějším úkolem bude implementace vyhlazení čáry pro chvějící se ruku, což je v současné době největším nedostatkem programu, jelikož při kreslení jsou veškeré „vibrace“ prstu přenášeny na kreslicí plátno.
- Dále by bylo vhodné zpřesnit výběr nástrojů z pracovního panelu, protože při prudkém, respektive trochu rychlejším, pohybu gesta swipe dochází k přeskokování nástrojů o více než jeden, jelikož zařízení ztrácí fokus a z jednoho gesta často vytvoří více.
- Rovněž by bylo vhodné vytvořit ovládací gest pro složení barev z jednotlivých kanálů RGB či CMYK, čímž by se eliminovalo použití klávesnice, respektive myši, a nastavování barev by tak záviselo pouze na Leap motionu.
- V neposlední řadě nelze opominout vytvoření gesta pro vrácení poslední změny v kompozici, nejlépe pomocí mávnutí na jednu stranu pro navrácení zpět a mávnutím na stranu druhou pro vrácení vpřed. To by byl další krok na cestě k úplné eliminaci jiných ovládacích periférií z aplikace.

4 Implementace

Tato kapitole se bude zabývat popisem vývoje aplikace ovládané zařízením Leap motion s využitím vývojového prostředí QT Creator v jazyce C++.

4.1 Vývoj programu

Vývoj byl rozdělen do několika fází, které odpovídají vodopádové metodě.

Jako první byla fáze **požadavků**. V ní bylo zapotřebí specifikovat požadavky na řešení. To znamená zjistit, co se od aplikace očekává, jaké nástroje jsou žádoucí, na jakou cílovou skupinu je program zaměřen.

Poté následuje fáze **návrh**, jež byla zaměřená na navrhnutí uživatelského rozhraní, rozložení okna aplikace a návrhu ovládání.

Dále se pokračuje fází **implementace**, v níž se na základě návrhu došlo k naprogramování veškerých funkcí a k doladění jejich vzájemného propojení.

Testování je zřejmě nejdůležitější a časově náročná fáze vývoje, kdy bylo nutné otestovat aplikaci v reálném provozu na skutečných uživateli, v mém případě na spolužácích, kamarádech a členech rodiny všech věkových kategorií.

Ve fázi **ladění** bylo potřeba na základě výsledků testování doladit aplikaci a opravit drobné chyby a nedostatky, jež byly nalezeny a k nimž při testování docházelo.

4.2 Kreslicí editor

Kreslicí editor je aplikace, jež využívá možnosti Leap motionu pro hravé kreslení na počítači za pomoci rukou, prstů a jejich gest. Okno aplikace bylo implementováno podle návrhu v 3.3.

Kreslicí plátno

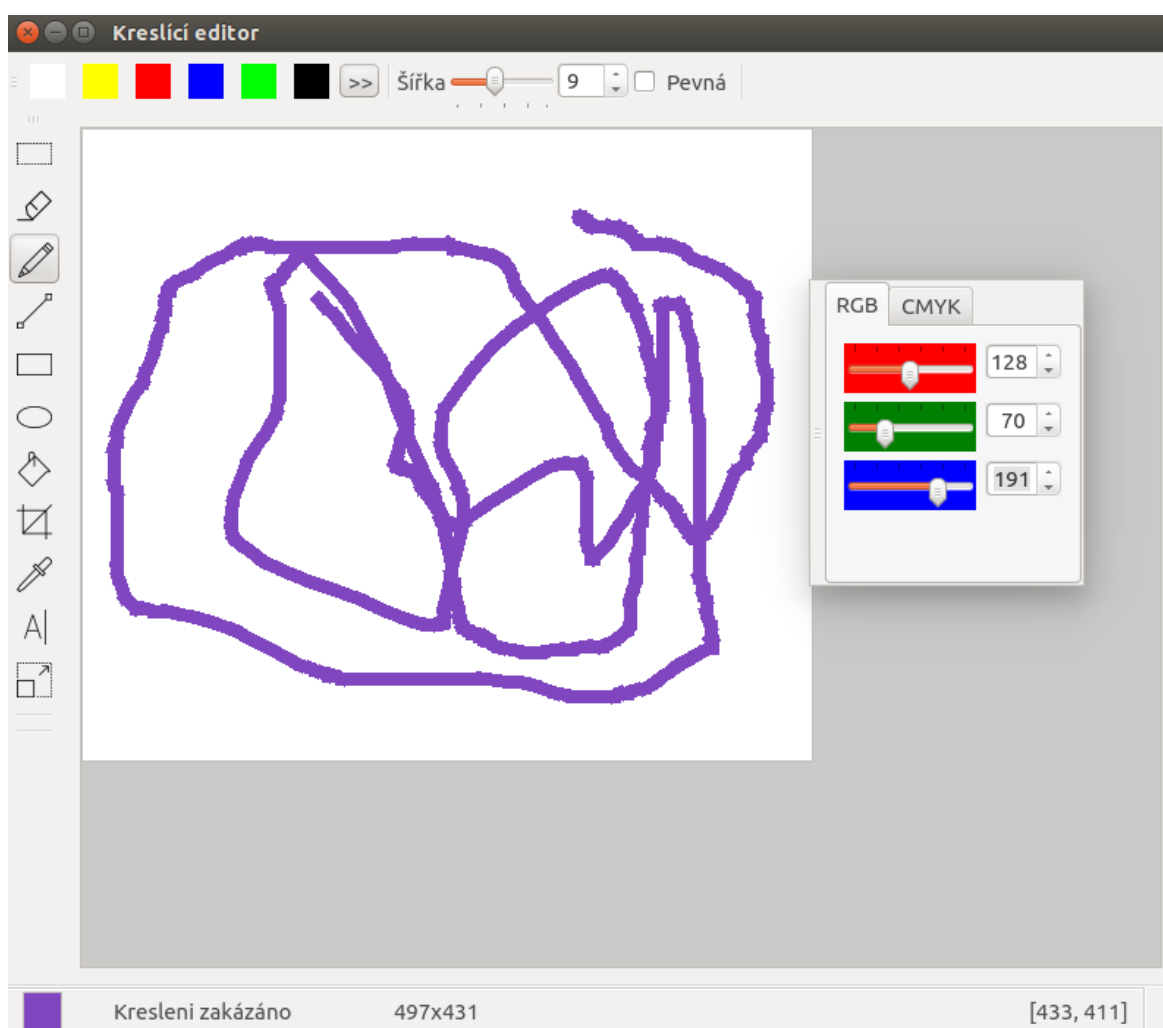
Kreslicí plátno je dominantní prvek, který zabírá většinu plochy okna. Účelem bylo vyvarování se situace, jež nastala v programu Malování Windows 7 a novějších, kde velikost kreslicího plátna ustoupila na úkor ribbonu (Microsoftem patentovaný ovládací prvek) s velkým množstvím nejrůznějších nástrojů. Velké plátno nabízí přehled nad dílem, aniž by byl uživatel nucen použít posuvník.

Panel nástrojů

Panel nástrojů je rozdělen na dvě části – kreslicí nástroje a základní barvy. Po spuštění se panel nachází na levé straně okna aplikace. Obě části panelu jsou nezávislé a mohou být libovolně přesouvány. Je tedy možné panel přichytit na jinou hranu okna, respektive nechat volně v prostoru.

Informační lišta

Informační lišta se nachází na spodní straně programu a její pozice je pevná. Na liště jsou zachyceny tři základní informace – aktuální barva, velikost kreslicího plátna, aktuální poloha kurzoru myši.



Obrázek 25: Kreslicí editor

4.3 Spojení s Leap motion

V této části kapitoly jsou popsány jednotlivé nástroje, funkce a jejich implementace. Rovněž je zde uveden způsob inicializace a komunikace s kontrolérem Leap motion a přepočítání souřadnic zařízení na velikost obrazovky monitoru.

Připojení Leap motion

Nejprve je nutné vytvořit kontrolér, aby bylo možné přistupovat k zařízení. Kontrolér je vytvořen v rámci hlavního okna aplikace. Ke kontroléru musí být přidán odposlouchávač (listener), který zpracovává příchozí události. Pro správnou funkčnost je nutné nakopírovat knihovny ke zdrojovým souborům a vše nainportovat do projektu.

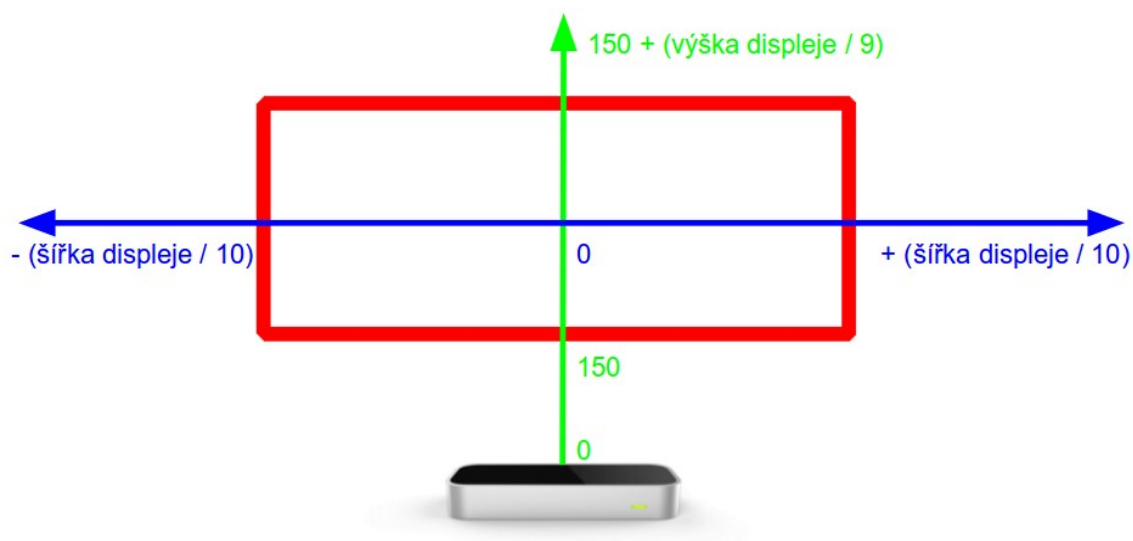
Jakmile jsou kontrolér a odposlouchávač vytvořeny, dojde k inicializaci zařízení a jeho napojení k aplikaci. To už se děje uvnitř odposlouchávače. Poté, v momentě dokončení inicializace, se zařízení zaostří a přizpůsobí okolí.

Po inicializaci a zaostření se z Leap motionu periodicky přijímají nové snímky a zpracovávají se informace ze získaných snímků. To zahrnuje cyklické procházení jednotlivých prstů na detekovaných rukou, aby se mohla zjistit potřebná informace.

Řízení kurzoru

Pohyb myši je řízen polohou ukazováčku pravé ruky. Špička ukazováčku se musí nacházet v pomyslném obdélníku, který se nachází ve výšce 150 mm nad zařízením a jehož velikost je závislá na rozlišení obrazovky. Z toho vyplývá, že souřadnice získané ze zařízení se nacházejí v intervalu, jehož mezní hodnoty na ose X odpovídají jedné desetině rozlišení displeje. Výška obdélníku je dána jednou devítinou výškou rozlišení obrazovky s počáteční hodnotou 150.

Jestliže se ukazováček nachází uvnitř obdélníku, musí se hodnoty přepočítat do intervalu, jenž má počátek v nule. Jakmile jsou hodnoty souřadnic upraveny, je následně s jejich pomocí vypočítána konečná poloha kurzoru, a to přičtením získaných hodnot k poslední poloze myši v rámci okna aplikace.



Obrázek 26: Znárodnění ovládacího prostoru

V případě, že se prst nenachází alespoň jednou souřadnicí ve znázorněném prostoru, je druhá souřadnice vypočtena stejným způsobem, avšak první hodnota je převzata z poslední pozice kurzoru. Pokud ani jedna z hodnot neleží ve vymezeném prostoru, je použita poloha myši z předcházejícího snímku.

4.4 Ovládání aplikace

Aplikace pro své ovládání využívá tři gesta:

- zahájení kreslení – spojení ukazováčku a prostředníčku
- výběr nástrojů a barev – gesto swipe
- zmenšení plátna – gesto pinch

Zahájení kreslení

Kreslení se zahájí, jestliže dojde ke spojení ukazováčku s prostředníčkem. Jak již bylo zmíněno, kurzor je ovládán pravým ukazováčkem, proto k němu stačí přiblížit prostředníček, čímž se započne kreslení. Tolerance pro přiblížení konečků prstů je padesát milimetrů ve všech osách.

```

//Ukazováček
if (finger.type() == 1)
{
    Vector pos = finger.tipPosition();    //pozice koncečku ukazováčku
    x = pos.x;
    y = pos.y;
    z = pos.z;
    w->moveFromListener(pos.x, pos.y, pos.z);    //předání polohy
}

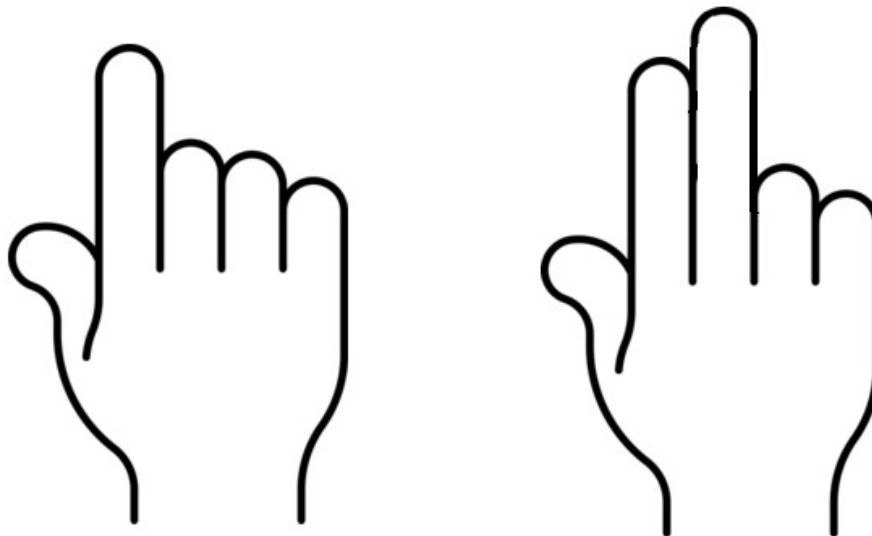
//Prostředníček
if (finger.type() == 2)
{
    Vector pos = finger.tipPosition();    //pozice koncečku prostředníčku
    double xx = pos.x - x;    //rozdíl ukazováčku a prostředníčku na X
    double yy = pos.y - y;    //rozdíl ukazováčku a prostředníčku na Y
    double zz = pos.z - z;    //rozdíl ukazováčku a prostředníčku na Z

    //pokud jsou koncečky od sebe ve vzdálenosti maximálně 50 mm ve všech osách
    if (xx > -25 && xx < 25 && yy > -25 && yy < 25 && zz > -25 && zz < 25)
        paintFinger = true;    //povolení kreslení
}

```

Algoritmus 1: Zahájení kreslení

Bylo důležité, aby kreslení bylo intuitivní a přechod mezi zahájením a ukončením kreslení byl plynulý a přirozený. V rámci testování vyšlo najevo, že není žádoucí kreslení řídit klávesnicí, případně druhou rukou, jelikož to mělo za následek zmatení uživatelů.



Obrázek 27: Znázornění ovládání – vlevo pouhé ovládání kurzoru, vpravo zahájení kreslení

Výběr nástrojů

Pro pohyb mezi nástroji je využito jedno z dostupných gest a to swipe. Gesto samotné bylo popsáno v Gesto Swipe, proto popíši jen použití. Šoupnutím doprava, respektive doleva, se vybírají jednotlivé nástroje. Šoupnutím nahoru, popřípadě dolů, je volena jedna ze základních šesti barev.

Změna velikosti plátna

Ke změně velikosti plátna je využito gesto pinch, tedy přiblížení, respektive oddálení, ukazováčku a palce pravé ruky. Ovšem aby gesto bylo aktivní, je nutné zvolit nástroj pro změnu plátna z nabídky panelu nástrojů. Tím se zabrání nežádoucímu změnění velikosti.



Obrázek 28: Znárodnění gesta pinch

4.5 Nástroje

Jak již bylo zmíněno, nástroje se nacházejí na panelu, který nemá fixní pozici a může být libovolně přesouván, jestliže uživateli nevyhovuje implicitní pozice při levém okraji aplikace. Editor poskytuje jedenáct různých nástrojů, které jsou realizovány pomocí tlačítek. U určitých nástrojů je možné zvolit dodatečnou funkci, například při kresbě obdélníku je nabídnuta možnost s výplní, či pouze obrysem tvaru.

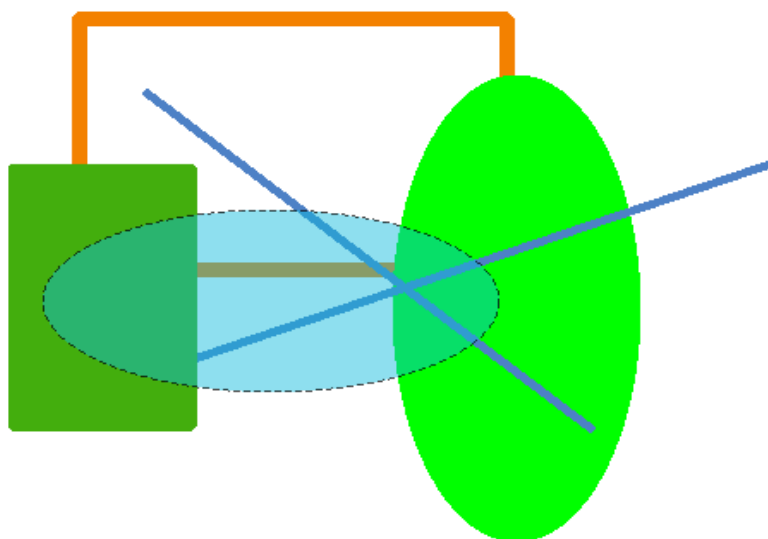


Obrázek 29: Panel nástrojů

Výběr oblasti

Výběr oblasti je první nástroj v nabídce. Jeho funkcí je vybrat část obrázku, která může být zkopírována do paměti nebo může být z kompozice úplně smazána. Nástroj je využit i v případě, jestliže se použije schránka operačního systému (clipboard) pro vložení obrázku z paměti. V tom případě vložený obrázek reprezentuje vybranou oblast, jež může být jakkoli přesouvána po plátně i mimo něj.

Vybraná oblast je zvýrazněna světle modrou barvou s přerušovaným obrysem. Barva je mírně průsvitná, aby původní plátno bylo stále dobře viditelné.



Obrázek 30: Elipsovitý výběr oblasti

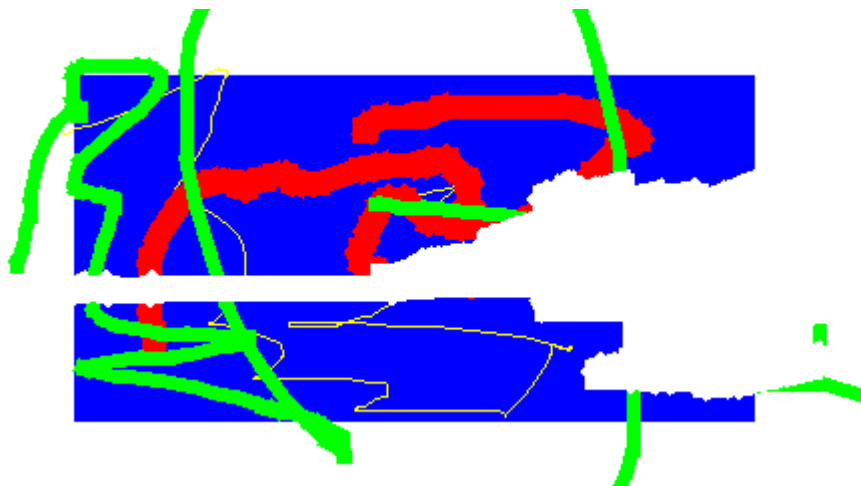
U nástroje je možné zvolit výběr ve tvaru obdélníku, či ve tvaru elipsy. Při kopírování elipsovitého výběru do paměti je ve skutečnosti kopírován obdélník opisující elipsu, jelikož operační systém nenabízí jiné řešení, a tak vnější strana elipsy je „malována“ průhlednou barvou. V případě kopírování elipsovitého výběru do jiných editorů, musí daný editor podporovat transparentní barvy.

Tužka a guma

Nástroj tužky je realizován kreslením krátkých úseček s využitím funkce *drawLine*. Pokud nakreslíme čáru, která je dlouhá deset pixelů, může se skládat ze dvou pětixelových úseček. Každá úsečka má počáteční a koncový bod. Počáteční bod je dán polohou kurzoru při zahájení kresby, koncový bod odpovídá aktualizované poloze kurzoru při jeho pohybu. Potom počátek bezprostředně následující úsečky odpovídá konci úsečky předchozí. Tímto způsobem můžeme nakreslit libovolný

tvár. Tato realizace kreslení s využitím úseček je prostředkově méně náročná oproti metodě kreslení po jednotlivých pixelech.

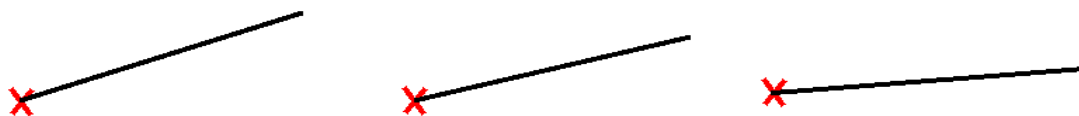
U mazání obsahu s využitím gumy se jedná o kreslení s tužkou s bílou barvou.



Obrázek 31: Znárodnění kresby tužkou a mazání gumou

Čára

Kreslení čáry je principiálně podobné jako u použití tužky, avšak čára se skládá z jedné úsečky, jejíž počáteční bod odpovídá zahájení kreslení a koncový bod aktuální poloze kurzoru. S pohybem myši rovněž dochází ke změně koncové souřadnice úsečky, která se průběžně překresluje, aby uživatel mohl vidět prozatímní výsledek, nicméně konečná podoba bude zaznamenána, jakmile uživatel přeruší kreslení.



Obrázek 32: Znárodnění kreslení čáry

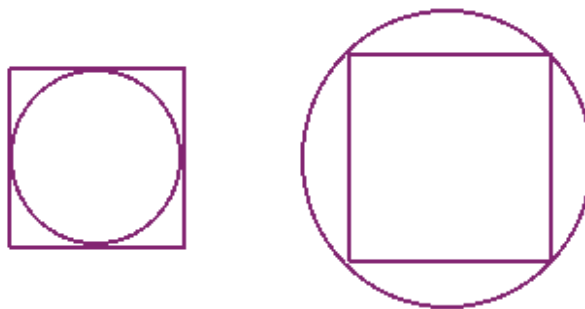
Obdélník, elipsa

Obdélník a elipsa jsou kresleny dostupnými funkcemi `drawRect` pro obdélník, respektive `drawEllipse` pro elipsu.

V obou případech se musí vytvořit obdélník, který je posléze vykreslen. Pro jeho sestavení je potřeba znát dva body – levý horní roh a pravý spodní roh. Výpočet délek stran se provede automaticky.

Umístění rohu není pevně vyžadováno, a není problém kreslit ve směru zprava doleva nebo zesponu nahoru. Z toho vyplývá, že jeden roh odpovídá poloze kurzoru při zahájení kreslení, druhý roh určuje pozice myši při přerušeni kreslení. Stejně, jako tomu je u kreslení čáry, tak i v tomto případě se objekt průběžně překresluje.

Jak již bylo zmíněno, i pro kresbu elipsy je potřeba sestrojít obdél ník, jenž bude určovat velikost elipsy. Výsledná elipsa bude do obdél níku vepsána.



Obrázek 33: Zobrazení čtverce a kruhu

Editor rovněž umožňuje kresbu rovnoměrných tvarů – čtverce a kruhu. Ty mohou být vytvořeny, pokud uživatel při kreslení drží klávesu CTRL. Potom se výška sestrojeného obdél níku převede na délku šířky, čímž vznikne čtverec.

```
QRect MainWindow::ctrlDrawingRect(QRect tmpRect)
{
    if (ctrlActive) //uživatel drží ctrl
    {
        int w = tmpRect.width();
        int h = tmpRect.height();
        if (h > 0 && w > 0) //směr kreslení obdél níku je doprava dolu
            tmpRect.setHeight(w);
        else if (h > 0 && w <= 0) //směr kreslení obdél níku je doleva dolu
            tmpRect.setHeight(-w);
        else if (h <= 0 && w > 0) //směr kreslení je doprava nahoru
            tmpRect.setHeight(-w);
        else if (h <= 0 && w <= 0) //směr kreslení je doleva nahoru
            tmpRect.setHeight(w);
    }
    return tmpRect; //vraceni původního obdél níku, nebo nového čtverce
}
```

Algoritmus 2: Kreslení čtverce

U obou nástrojů je možné zvolit, zdali se má tvar kreslit s výplní, nebo pouze obrysem. Jako výchozí volba nástroje je zvolena možnost kreslení obrysu útvaru.



Obrázek 34: Kresba s výplní a bez výplně

Výplň (plechovka barvy)

Plechovka barvy je nástroj pro vyplnění uzavřené oblasti zvolenou barvou. Nejdříve je potřeba zjistit, jestli nová barva, kterou chceme nanést, není náhodou totožná s tou, jež už je použita v místě stisku kurzoru. V případě shody se nic nestane.

```
//pixel z místa stisku kurzoru
QRgb pixel = img.pixel(startPos.x(), startPos.y());
QColor oldColor(pixel);

//porovnání barvy pixelu se zvolenou barvou
if (oldColor != actualColor)
    drawingFiller(startPos.x(), startPos.y(), pixel, actualColor.rgb(),
&img); //vyplnění oblasti
```

Algoritmus 3: Zahájení vyplňování

Při vyplňování oblasti novou barvou se postupně prochází plocha po řádcích s využitím funkce *drawingFiller*. Začíná se na souřadnicích pozice kurzoru a jde se směrem doleva. Každý pixel se porovnává, zda jeho barva je odlišná od zvolené barvy. V případě neshody se daný bod obarví novou barvou a pokračuje se na sousední pixel. Takto se pokračuje až na okraj plátna nebo do nalezení shody při porovnání barev. V obou situacích se tímto postupem zjistí levý okraj výplně daného řádku. Tento postup se poté opakuje i ve směru doprava, aby se našel pravý okraj.

```
while (true)
{
    if (img->pixel(tmpX, y) != oldColor) //shoda barvy
        break; //konec cyklu

    img->setPixel(tmpX, y, newColor); //nastavení nové barvy
    if (tmpX > 0) //nejsme na hraně
    {
        tmpX--; //počítadlo sloupce
        leftBorder = tmpX; //nastavení nové hrany
    }
    else
        break; //konec cyklu
}
```

Algoritmus 4: Nalezení hranic při vyplňování

Jakmile jsou zjištěny obě hrany určující rozsah zvoleného řádku, cyklicky se prochází zleva doprava a pro každý sloupec se volá funkce *drawingFiller*, kterou se posune o jeden řádek výš a posléze o jeden řádek níž. Tímto rekurzivním způsobem se vybarví celá oblast libovolného tvaru.

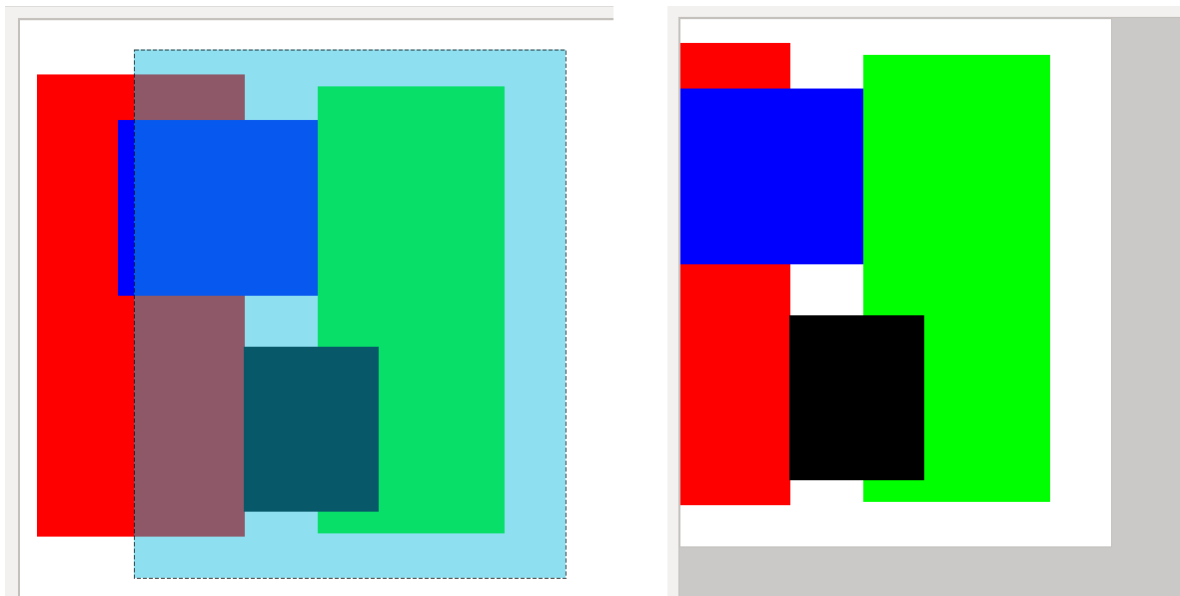
```
for (int newX = leftBorder; newX < rightBorder; newX++)//projdu řádek zleva
{
    if (y < 1 || y >= img->height()-1)           //nalezení výškové hrany
        break;                                   //konec cyklu

    QRgb currentColor = img->pixel(newX, y-1);    //zjištění barvy
    if (currentColor == oldColor && currentColor != newColor)//porov. barev
        drawingFiller(newX, y-1, oldColor, newColor, img); //rekurze
    currentColor = img->pixel(newX, y+1);        //zjištění barvy
    if (currentColor == oldColor && currentColor != newColor)//porov. barev
        drawingFiller(newX, y+1, oldColor, newColor, img); //rekurze
}
```

Algoritmus 5: Překreslení pixelů

Ořezání

Jak název napovídá, nástroj slouží k provedení výřezu části obrázku z plátna. K realizaci se využívá nástroj pro výběr, kterým se určí oblast, jež se má v díle zachovat. Ořezání se provede ihned poté, co je ukončeno gesto, jímž se vybrala patřičná část obrázku. Jakmile je ořez vykonán, přetransformují se rozměry plátna tak, aby odpovídaly velikosti výřezu, který je přesunut do levého horního rohu aplikace.



Obrázek 35: Ořezání obrázku: vlevo výběr oblasti, vpravo provedení ořezání

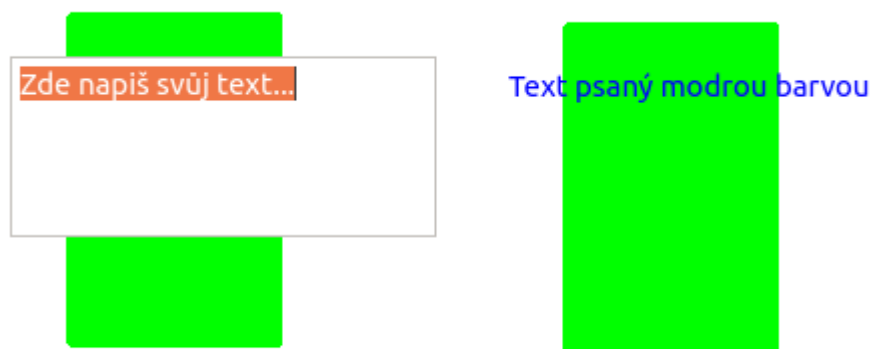
Kapátko

Tento nástroj slouží ke zjištění barvy pixelu na plátně v místě pozice kurzoru. Jakmile je detekováno gesto kreslení, přečte se barva z RGB složek pixelu a tyto hodnoty se nastaví pro aktuální barvu. Tímto způsobem je možné zjistit složení jakékoli barvy a opětovně ji v kompozici použít. Detailnější popis práce s RGB je k nalezení v 4.6.

Vložení textu

Pro vložení textu je nutné, aby uživatel použil hardwarovou klávesnici, jelikož se mi nezdálo příliš vhodné řešení s použitím virtuální klávesnice. Rovněž se domnívám, že se nejedná o stěžejní funkci, která by byla využívána s velkou pravidelností, tudíž by uživateli v těchto případech nemuselo vadit psání textu bez Leap motinu, ale naopak by to mohl ocenit.

Nejprve uživatel vytvoří textové pole, které se zvětšuje/zmenšuje v závislosti na pohybu kurzoru. Když je textové pole vloženo, označí se výchozí text, aby došlo k jeho smazání, jakmile uživatel začne psát svůj text. Až je text napsán, kliknutí mimo pole způsobí uložení textu do obrázku. Barva textu odpovídá barvě aktuálně nastavené.



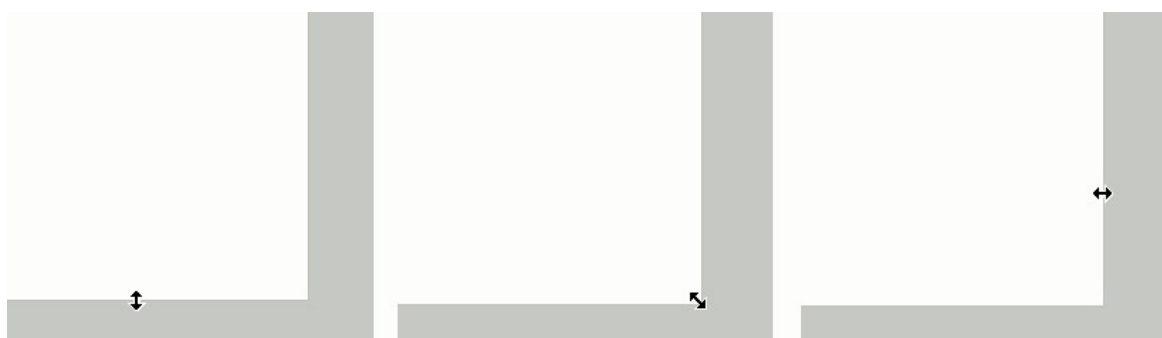
Obrázek 36: Vložení textu

Pokud není změněno znění výchozího textu, nedojde k jeho uložení, ale považuje se to za „zrušení“ akce. Editor neposkytuje uživateli možnost dodatečné úpravy vloženého textu, který se už stal pevnou součástí obrázku. Dále není nabídnuta příležitost ke změnám vlastností textu, jako jsou například typ či velikost fontu.

Změna rozměrů plátna

Změnit velikost plátna je možné dvěma způsoby – buď využitím kurzoru myši, nebo gestem pinch. Nicméně v obou případech je nutné zvolit nástroj k tomu určený, aby se zabránilo nežádoucím změnám v průběhu práce.

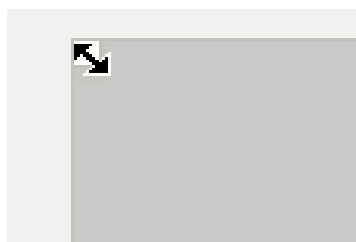
V prvním případě je nutné přesunout kurzor na pravou nebo spodní hranu plátna, případně do jeho pravého spodního rohu. Jakmile se kurzor myši nachází ve správné pozici, která má jistou míru tolerance, dojde ke změně ukazatele na dvojitou šipku, jež indikuje možnost zahájení činnosti změny rozměrů.



Obrázek 37: Změna rozměrů plátna: výška, výška i šířka, šířka

Druhou metodou je využití gesta pinch a tím měnit rozměry plátna v závislosti na vzdálenosti ukazováčku a palce pravé ruky. Gesto má nevyhnutelně omezen pracovní prostor, tudíž je aktivní pouze v případě, že se prsty nacházejí za nultým bodem souřadnicové osy Z, aby bylo snazší gesto použít opakovaně za sebou pro dosažení požadované změny pro velké rozdíly velikostí. Samozřejmě je nutné podotknout, že tato varianta změny rozměrů není v současné podobě příliš přesná.

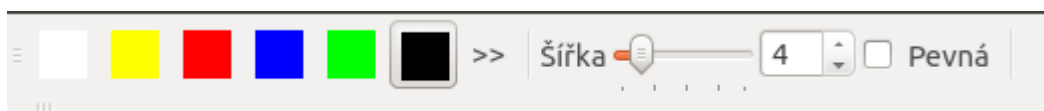
Minimální velikost plátna je nastavena na deset pixelů na šířku a deset pixelu na výšku. Za tuto hranici již zmenšovat nelze.



Obrázek 38: Minimální velikost plátna

4.6 Panel barev

Jak již bylo zmíněno, uživateli je nabídnuto šest základních barev a to bílá, žlutá, červená, modrá, zelená a černá, mezi nimiž může přecházet gestem švihnutí směrem nahoru, případně dolů. Zvolená barva je zvýrazněna označením tlačítka, a rovněž dojde ke změně barvy informační ikony, jež se nalézá na informačním panelu.



Obrázek 39: Panel barev

Jestliže uživateli nestačí nabídnuté základní barvy, může zvolit nástroj kapátko, pomocí kterého lze vybrat jakýkoliv pixel obrázku, čímž se nastaví jeho barva. Více o kapátku v kapitole 4.5.

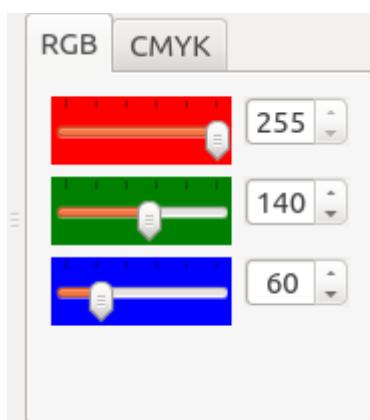
V případě, že uživatel nemá v kompozici vhodnou barvu, aby využil funkce kapátka, může si navolit barvu pomocí jednotlivých složek. Uživatel možnosti pro nastavení barvy zpřístupní kliknutím na tlačítko znázorňující dvě šipky. Bohužel, toto nastavení je závislé na myši.



Obrázek 40: Otevření nabídky pro možnosti barev

Palety barev

Aplikace nabízí dvě různé palety barev a to RGB a CMYK. V RGB variantě lze cílovou barvu sestavit nastavením jednotlivých složek RGB, tedy červené (Red), zelené (Green) a modré (Blue).



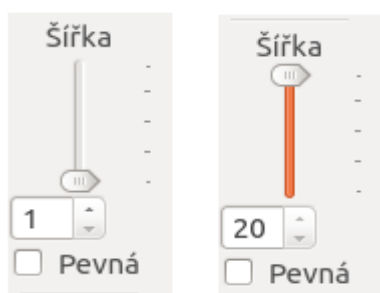
Obrázek 41: Nastavení RGB složek

Hodnoty dílčích kanálů je možné určit číslem v rozmezí 0-255. V paletě CMYK se barva nastavuje obdobně, jen s tím rozdílem, že se využívají složky azurová (Cyan), purpurová (Magenta), žlutá (Yellow) a černá (Key).

Jednou z variant nastavení barvy je vepsání čísla rovnou do pole. Další možností, jak nastavit výslednou barvu, je využití posuvníku. Poslední způsob, kterým lze výslednou barvu doladit, je s využitím šipek, jež se nacházejí bezprostředně vedle čísla, kterými lze hodnotu postupně zvětšovat, případně zmenšovat, o jedničku. Samozřejmě hodnota čísla odpovídá hodnotě posuvníku a naopak.

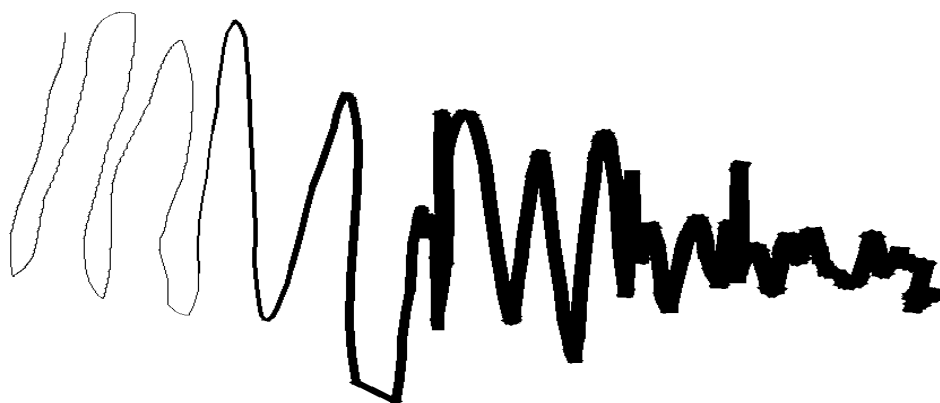
Šířka čáry

Poslední ovládací prvek, který se nachází na panelu s barvami, je ovládání, respektive indikace, šířky čáry. Šířka může nabývat hodnoty v rozsahu jeden až dvacet pixelů.



Obrázek 42: Nastavení/indikace šířky čáry

Primárně je šířka čáry ovládána pohybem kreslicí ruky po ose Z. Čím více se uživatelův prst přibližuje obrazovce monitoru, to znamená pohybuje se hlouběji, tím více se čára stává širší.



Obrázek 43: Přechod mezi tenkou a hrubou čárou

Jestliže uživatel vyžaduje pevnou tloušťku čáry, je mu poskytnuto zaškrtnuté tlačítko, kterým se zamezí ovládání šířky pohybem prstu, a po celou dobu kreslení tloušťka čáry odpovídá zvolené

hodnotě, jež lze nastavit posuvníkem, případně vepsat číselnou hodnotu do pole. Rovněž i zde hodnota čísla odpovídá hodnotě posuvníku a naopak.



Obrázek 44: Pevná šířka čáry

4.7 Informační lišta

Informační lišta obsahuje základní informace důležité pro uživatele.



Obrázek 45: Informační lišta

Aktuální barva

Jedná se o ikonu, která je umístěna v levém spodním rohu programu. Ikona indikuje právě zvolenou barvu, ať už je zvolena z nabídky základních barev nebo se jedná o barvu poskládanou z jednotlivých složek palet RGB či CMYK. Aktualizace indikátoru probíhá okamžitě, a tedy nevzniká žádná prodleva při skládání barev.

Povolené kreslení

Text „Kreslení zakázáno/povoleno“ informuje uživatele o tom, zda je mu v dané chvíli kreslení umožněno. Text se mění automaticky v závislosti na gestu pro zahájení kreslení.

Velikost plátna

Indikace rozměrů plátna je ve formě čísla a nachází se zhruba uprostřed informační lišty. Velikost je dána šířkou a výškou plátna. Při změně rozlišení plátna nedochází k žádnému zpoždění při zobrazení aktuálních hodnot.

Pozice kurzoru

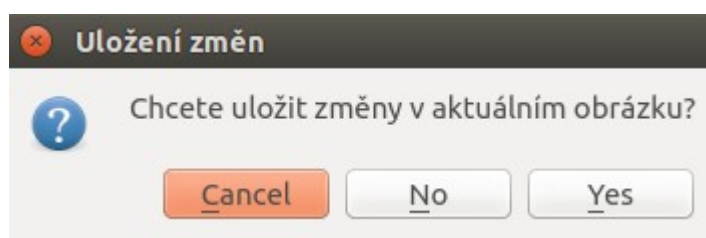
Pozice kurzoru je dána souřadnicemi X a Y v rámci plátna. To znamená, že nultá souřadnice se nachází v levém horním rohu plátna a maximální hodnoty odpovídají šířce, respektive výšce, plátna v pravém spodním rohu.

4.8 Hlavní nabídka

Hlavní nabídka obsahuje dvě záložky, jednu pro práci se soubory, druhou pro práci v kompozici.

Soubor

Z nabídky je možné vytvořit nový obrázek a to s velikostí plátna 500 x 500 pixelů. Další volbou je otevření již existujícího obrázku nebo fotky. Podporovány jsou různé formáty, počínajíc JPEG, přes BMP, konče PNG. Celkově je podporováno deset různých formátů. U obou voleb je uživatel dotázán, zdali chce uložit dosavadní práci v díle.



Obrázek 46: Dialog pro uložení

Poslední volba je uložení kompozice do souboru. Výchozí formát souboru je „Portable Network Graphics“, ale uživatel může sám zadat svůj formát, který požaduje. Rovněž i zde je podporováno deset různých formátů.

Výchozí umístění je nastaveno na standardní složku s obrázky daného operačního systému. Pokud uživatel nezadá název, zvolí se pojmenování „image“. V případě, že uživatel odešle prázdné pole, soubor se uloží pod názvem „image.png“.

Úpravy

V záložce úpravy se nacházejí funkce pro základní úpravy kompozice. Nejdůležitějšími položkami jsou funkce zpět a vpřed. V paměti je uloženo posledních dvacet změn, které mohou být navráceny.

Dále se zde nacházejí klasické funkce jako kopírování, vložení, vyjmutí, smazání a označení celého plátna.

Pokud chce uživatel použít funkci smazání, musí vybrat oblast, jinak dojde k „vyčištění“ celého plátna. V případě nástrojů kopírování a vyjmutí musí rovněž vybraná oblast, jinak se neprovede žádná akce. Jestliže uživatel bude vkládat obrázek, automaticky se změní vybraný nástroj na „výběr oblasti“ s rozměry, jež odpovídají vkládanému obrázku, a tak je umožněno s ním pohybovat a libovolně jej umístit. Samozřejmě všechny nástroje a funkce mají klávesovou zkratku, pro snadnější ovládání při využití klávesnice.

5 Testování

Testování probíhalo ve dvou hlavních fázích. V první fázi, do které se zapojilo pár vyvolených, bylo účelem zjistit, jestli je navržené ovládání vhodné a jeho použití intuitivní. Ve druhé fázi, v níž už spolupracovalo výrazně více účastníků, šlo o testování aplikace jako celku.

Všechny testovací běhy probíhaly na notebooku Asus G55VW s 15,6“ full HD displejem v systému Ubuntu 14,04 LTS 64bitové verze. I když editor díky multiplatformnímu vývoji umožňuje běh i na systémech MS Windows, nebyla tato verze dostatečně odzkoušena v porovnání s verzí pro Linux, na kterou jsem byl primárně zaměřený.

5.1 Alfa verze

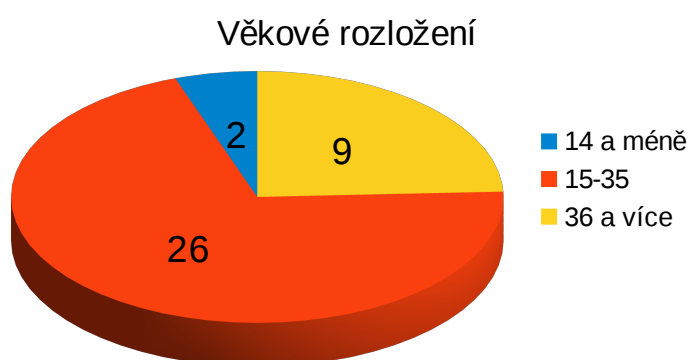
Testování alfa verze kreslicího editoru, tedy první fáze, probíhalo v uzavřeném okruhu několika mých známých, kteří mi pomohli zhodnotit dosavadní stav editoru a klasifikovat zvolená gesta pro ovládání aplikace. Testování probíhalo ve třech etapách.

- V první etapě testování editoru se kreslení zahajovalo, respektive ukončovalo, s využitím klávesnice. Jak z testování vyplynulo, nebyla to nejlepší volba, jelikož se vytratila nezávislost Leap motionu. Proto bylo zapotřebí vymyslet nový způsob, jímž by se dalo kreslení ovládat.
- Druhá etapa testovala první navržený způsob, kterým bylo využití obou rukou s tím, že u kreslení samotného by se nic nezměnilo a stále by se využívala pravá ruka, avšak zahajování a přerušení kreslení by se ovládalo pohybem levé ruky. Ovšem tato realizace přinášela komplikace, jelikož zařízení mělo problém se správnou detekcí končetin při jejich křížení. To způsobovalo výpadky a přerušování kreslení, proto bylo od tohoto řešení ustoupeno.
- Poslední etapa alfa testování se zabírala druhým způsobem, jež byl pro řízení kreslení navržen, a stával se z jednoduchého pohybu dvou prstů. Kreslení se zahájí ve chvíli, kdy se ukazováček a prostředníček přiblíží dostatečně blízko k sobě. Více v části 4.4. Gesto vychází z předpokladu, že žádný člověk nebude mít problém s oddělováním a opětovným spojováním ukazováčku s prostředníčkem, a tento odhad se potvrdil při testování třetí alfa verze programu.

5.2 Beta verze

Testování beta verze programu probíhalo v širší skupině uchazečů. Konečný počet se zastavil na třiceti sedmi respondentech a každý z nich vyplnil dotazník, v němž sdělil své pocity z programu.

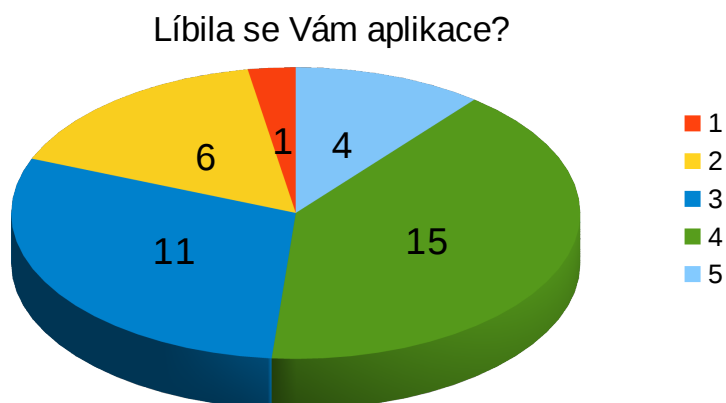
Přestože se jednalo o členy rodiny, spolužáky, či známe, snažil jsem se pokrýt všechny věkové kategorie. Nejvíce byla zastoupena skupina ve věku 15 až 35 let, přesněji necelými jednadsmdesáti procenty. Ze skupiny 36 let a starší se účastnilo devět respondentů a zbylí dva testeři byli mladší patnácti let.



Obrázek 47: Graf věkového rozložení testujících

5.3 Zhodnocení testování

Z tohoto testování vyšlo najevo, že většina uživatelů byla s aplikací spokojená, a hodnotili aplikaci kladně. Necelých čtyřicet jedna procent testujících, což odpovídá patnácti lidem, hodnotili program čtyřmi body z pěti. Průměrným hodnocením, tedy třemi body, klasifikovalo necelých třicet procent respondentů. Nejnižší známku zanechal jeden testující.



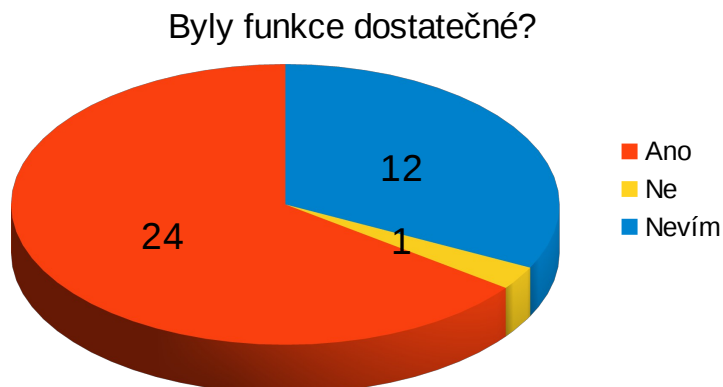
Obrázek 48: Graf hodnocení aplikace

Důležitým bodem dotazníku byla otázka, zdali je navržené ovládání intuitivní. Jak z odpovědí vyplývá, skoro tři čtvrtiny respondentů byli s ovládáním spokojeni a považovali jej za vhodné a dostatečně intuitivní.



Obrázek 49: Graf intuitivnosti ovládní

Rovněž většina dotázaných považovala nabízené funkce jako dostatečné a až na pár případů, v níž by přivítali možnost nastavení přesných rozměrů plátna, nepostrádali žádný nástroj.



Obrázek 50: Graf dostatečnosti nabízených funkcí

Z výsledků dotazníků a osobních rozhovorů vyplývá, že kreslicí editor je dostatečně kvalitní program, který je schopný obstát na trhu.

6 Závěr

Cílem práce bylo nastudovat možnosti bezdotykového ovládání a navrhnout vhodný program, který by získané znalosti realizoval. Práce se zaměřila na popsání technologie Leap motion a její použití v aplikaci kreslicí editor. Cíl práce byl splněn.

Prostudoval jsem dostupnou literaturu na téma bezdotykového ovládání a interakce ve 3D prostředí a výsledky jsou shrnuty v druhé kapitole. Navrhnuté principy interakce jsou předmětem první poloviny třetí kapitoly. Navrhnutý způsob implementace je shrnutý v druhé části třetí kapitoly. Implementace softwaru je realizována ve čtvrté kapitole a demonstrace jeho funkčnosti je popsána v kapitole pět.

Z testování, popsaného v páté kapitole práce, vyplývá, že pro 73 procent uživatelů se kreslicí editor jeví jako dostatečně intuitivní a přehledný. Rovněž nabízené funkce byly dostatečné pro 24 uživatelů z celkového počtu 37 testujících. Co se týče celkového hodnocení aplikace, průměrný počet bodu je 3,4 z 5 možných.

V budoucí práci by bylo vhodné vytvořit další gesta pro ovládání aplikace, zvláště i u méně podstatných funkcí, což by vedlo k eliminaci používání klávesnice a myši v aplikaci. Rovněž by bylo žádoucí lépe pracovat s citlivostí zařízení, tedy implementovat vyhlazení čáry pro kreslení a také přesněji vybírat nástroje z panelu.

Práce mi pomohla dostat se hlouběji do problematiky bezdotykového ovládání, jež je, dle mého názoru, do budoucna důležitá, zvláště z pohledu využití virtuální reality a v serióznějším pojetí pro hendikepované lidi, pro které to může být jediný způsob, jak ovládat počítač.

Literatura

- [1] *Leap Motion. Wikipedia* [online]. 2016, last modified on 2 February 2016 at 10:23. [cit. 2016-2-5]. Dostupné z: https://en.wikipedia.org/wiki/Leap_Motion
- [2] JAVŮREK, K.: *Leap Motion: revoluční ovládání na vlastní (ne)dotek* [online]. 2013. [cit. 2016-2-5]. Dostupné z: www.zive.cz/clanky/leap-motion-revolucni-ovladani-na-vlastni-dotek/budoucnost-a-spravne-pouziti/sc-3-a-169030-ch-87008/default.aspx
- [3] PETŘÍK, J.: *Skokem k zábavě aneb vzestupy a pády herní periferie Kinect* [online]. 2015. [cit. 2016-2-5]. Dostupné z: <http://doupe.zive.cz/clanek/skokem-k-zabave-aneb-vzestupy-a-pady-herne-periferie-kinect>
- [4] *Xbox One. Wikipedia* [online]. 2016, naposledy editována 11. 1. 2016 v 23:43. [cit. 2016-2-5]. Dostupné z: https://cs.wikipedia.org/wiki/Xbox_One#Kinect
- [5] *Tobii Technology, Wikipedia* [online]. 2016, last modified on 21 January 2016, at 16:46. [cit. 2016-2-5]. Dostupné z: https://en.wikipedia.org/wiki/Tobii_Technology
- [6] HOLČÍK, T.: *Tobii Gaze: ovládání počítače očima funguje [CES]* [online]. 2013. [cit. 2016-2-5]. Dostupné z: <http://www.zive.cz/clanky/tobii-gaze-ovladani-pocitace-ocima-funguje-ces/sc-3-a-167110/default.aspx>
- [7] JAVŮREK, K.: *MYO: tohle je možná budoucnost ovládání gesty* [online]. 2013. [cit. 2016-2-5]. Dostupné z: <http://www.zive.cz/clanky/myo-tohle-je-mozna-budoucnost-ovladani-gesty/sc-3-a-167757/default.aspx>
- [8] System architecture [online]. [cit. 2016-5-15]. Dostupné z: https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Architecture.html
- [9] *C++ SDK Documentation* [online]. [cit. 2016-2-5]. Dostupné z: <https://developer.leapmotion.com/documentation/cpp/index.html>
- [10] *API overview* [online]. [cit. 2016-2-5]. Dostupné z: https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Overview.html

- [11] SPIEGELMOCK, M.: *Leap Motion Development Essentials*. Packt Publishing, První vydání, 2013, ISBN 978-1-84969-772-9.
- [12] Leap Motion with Orion [online]. [cit.2016-3-28].
Dostupné z: <https://www.leapmotion.com/product/vr>
- [13] *Introducing Orion: Next-Generation Hand Tracking for VR* [online]. 2016. [cit. 2016-3-28].
Dostupné z: <http://blog.leapmotion.com/orion/>
- [14] HUTCHINSON, L.: *Leap Motion gets a head-mounted upgrade, becomes a VR controller* [online]. 2016. [cit. 2016-3-28]. Dostupné z: <http://arstechnica.com/gadgets/2014/08/leap-motion-gets-a-head-mounted-upgrade-becomes-a-vr-controller/>
- [15] ROBERTSON, A.: *Leap Motion's revamped hand tracking is getting built straight into VR headsets* [online]. 2016. [cit. 2016-3-28]. Dostupné z: <http://www.theverge.com/2016/2/17/11021214/leap-motion-hand-tracker-virtual-reality-orion-mobile-vr>
- [16] COLGAN, A.: *Power at Your Fingertips: Pinch Utilities for Orion* [online]. 2016. [cit. 2016-5-15]. Dostupné z: <http://blog.leapmotion.com/power-fingertips-pinch-utilities-orion/>
- [17] *Cube simulator* [online]. [cit. 2016-5-15].
Dostupné z: <https://developer.leapmotion.com/gallery/cube-simulator>
- [18] *Hand Capture* [online]. [cit. 2016-5-15].
Dostupné z: <https://developer.leapmotion.com/gallery/hand-capture>
- [19] *Pinch Draw* [online]. [cit. 2016-5-15].
Dostupné z: <https://developer.leapmotion.com/gallery/pinch-draw>

Seznam obrázků

Obrázek 1: Leap motion.....	3
Obrázek 2: Kinect.....	4
Obrázek 3: Tobii EyeX.....	4
Obrázek 4: MYO.....	5
Obrázek 5: Native interface.....	6
Obrázek 6: WebSocket interface.....	7
Obrázek 7: Podoba Visualizeru.....	8
Obrázek 8: Pravotočivý souřadnicový systém.....	9
Obrázek 9: Orientace ruky je definována normálovým vektorem dlaně a jejím směrem.....	10
Obrázek 10: Pozice a směr je určen konečky prstů.....	11
Obrázek 11: Circle - prst krouží po kružnici.....	11
Obrázek 12: Swipe - dlouhý lineární pohyb ruky a prstů.....	12
Obrázek 13: Key tap - simulace stisku tlačítka.....	13
Obrázek 14: Screen tap - simulace klepnutí na obrazovku.....	13
Obrázek 15: Vlastnosti pohybu.....	14
Obrázek 16: Snímek senzoru.....	14
Obrázek 17: Leap motion ve spojení s Oculus riftem.....	15
Obrázek 18: Přichycení Leap motion k Oculus riftu.....	16
Obrázek 19: Zorná pole Leap motinu a Oculus riftu.....	16
Obrázek 20: Pinch Utilities.....	17
Obrázek 21: Cube simulator.....	18
Obrázek 22: Hand Capture.....	18
Obrázek 23: Pinch Draw.....	19
Obrázek 24: Návrh okna aplikace.....	21
Obrázek 25: Kreslící editor.....	24
Obrázek 26: Znáznění ovládacího prostoru.....	26
Obrázek 27: Znáznění ovládní – vlevo pouhé ovládní kurzoru, vpravo zahájení kreslení.....	27
Obrázek 28: Znáznění gesta pinch.....	28
Obrázek 29: Panel nástrojů.....	28
Obrázek 30: Elipsovité výběr oblasti.....	29
Obrázek 31: Znáznění kresby tužkou a mazání gumou.....	30
Obrázek 32: Znáznění kreslení čáry.....	30

Obrázek 33: Zobrazení čtverce a kruhu.....	31
Obrázek 34: Kresba s výplní a bez výplně.....	32
Obrázek 35: Ořezání obrázku: vlevo výběr oblasti, vpravo provedení ořezání.....	33
Obrázek 36: Vložení textu.....	34
Obrázek 37: Změna rozměrů plátna: výška, výška i šířka, šířka.....	35
Obrázek 38: Minimální velikost plátna.....	35
Obrázek 39: Panel barev.....	36
Obrázek 40: Otevření nabídky pro možnosti barev.....	36
Obrázek 41: Nastavení RGB složek.....	36
Obrázek 42: Nastavení/indikace šířky čáry.....	37
Obrázek 43: Přejít mezi tenkou a hrubou čárou.....	37
Obrázek 44: Pevná šířka čáry.....	38
Obrázek 45: Informační lišta.....	38
Obrázek 46: Dialog pro uložení.....	39
Obrázek 47: Graf věkového rozložení testujících.....	41
Obrázek 48: Graf hodnocení aplikace.....	41
Obrázek 49: Graf intuitivnosti ovládání.....	42
Obrázek 50: Graf dostatečnosti nabízených funkcí.....	42

Seznam algoritmů

Algoritmus 1: Zahájení kreslení.....	27
Algoritmus 2: Kreslení čtverce.....	31
Algoritmus 3: Zahájení vyplňování.....	32
Algoritmus 4: Nalezení hranic při vyplňování.....	32
Algoritmus 5: Překreslení pixelů.....	33

Příloha A

A.1 Dotazník

Kreslící editor pro Leap motion

Formulář k programu Kreslící editor s Leap motion

*Povinné pole

Líbila se Vám aplikace? *

1 2 3 4 5

Ne Ano

Bylo ovládání intuitivní? *

- Ano
 Ne
 Nevím

Změnili byste způsob ovládání? *

- Ano
 Ne
 Nevím

Byly funkce dostatečné? *

- Ano
 Ne
 Nevím

Používali byste editor i nadále? *

- Ano - Windows Vista a starší
 Ano - Windows 7 a novější
 Ano - jiné
 Nevím
 Ne

Chybí Vám nějaký nástroj? *

- Ano
- Ne
- Nevím

Pokud Vám nějaký nástroj chybí, jaký?

Uvedte Váš věk. *

- 14 a méně
- 15-35
- 36 a více

Odeslat

Nikdy přes Formuláře Google neposílejte hesla.

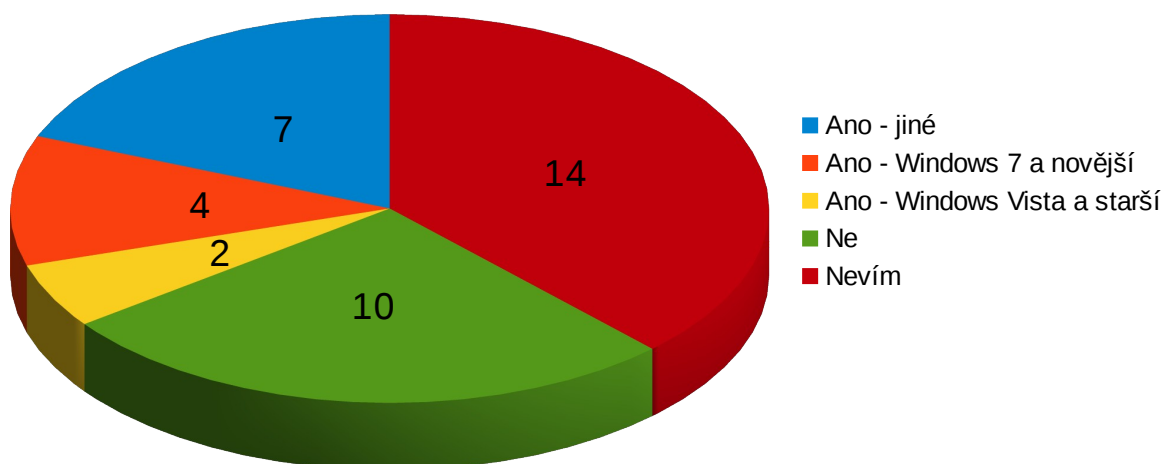
Adresa dotazníku: <http://goo.gl/forms/Ae0BWVRolR>

A.2 Vyhodnocení dotazníku

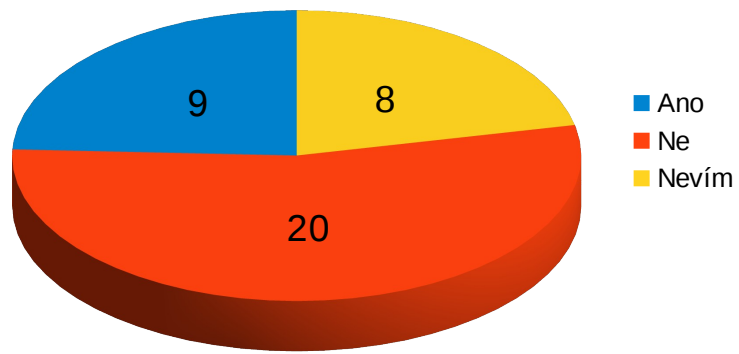
Počet respondentů: 37

Grafy otázek, které nebyly zmíněny v 5.3.

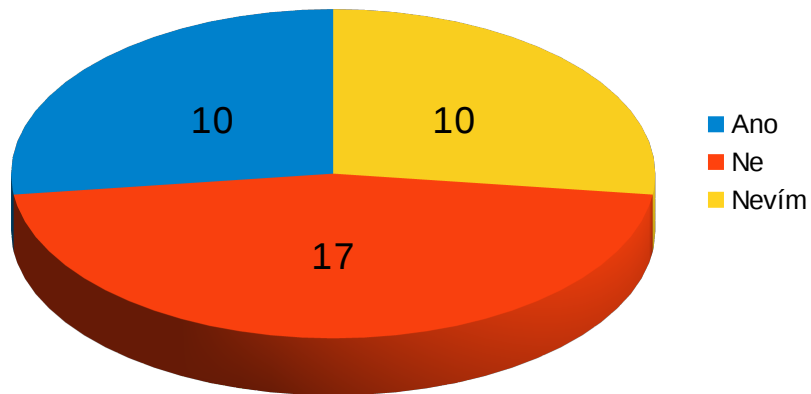
Používali byste editor i nadále ve Vašem systému?



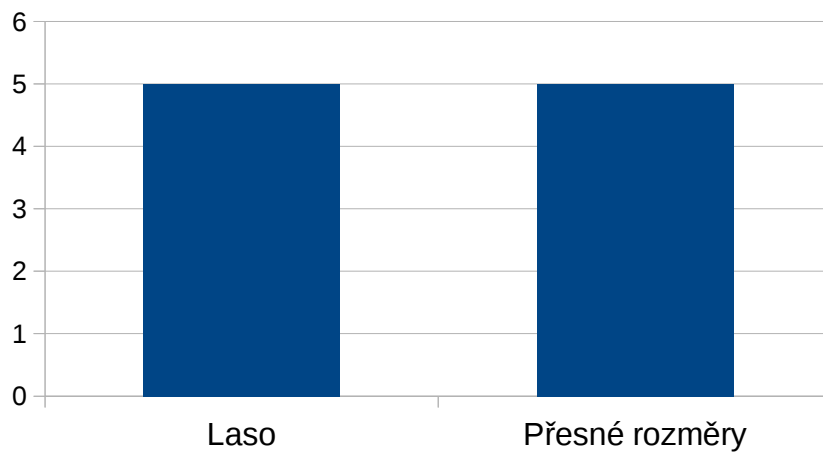
Změnili byste způsob ovládání?



Chybí Vám nějaký nástroj?



Chybí Vám nějaký nástroj?



Příloha B

B Obsah CD

K práci je přiložen optický disk, jenž obsahuje text bakalářské práce, dále zdrojové kódy k programu vytvořeném v rámci této práce ve verzi pro operační systém Windows i Linux.