

České vysoké učení technické v Praze  
Fakulta elektrotechnická

katedra počítačové grafiky a interakce

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Jan Starý**

Studijní program: Softwarové technologie a management  
Obor: Web a multimedia

Název tématu: **Emulace myši a klávesnice pro motoricky postižené uživatele**

Pokyny pro vypracování:


Vytvořte aplikaci, která bude emulovat myš pomocí sledování pozice uživateleovy hlavy a neverbálního hlasového vstupu. Zaměřte se na použitelnost výsledné aplikace pro uživatele s vážným postižením horních končetin. Pro zpracování sledování hlavy i neverbálního hlasového vstupu použijte existující knihovny. Dále navrhnete a implementujete prediktivní virtuální klávesnici, která bude přizpůsobená tomuto ovládání. Půžitelnost aplikací ověřte v testu s uživateli se simulovaným postižením.

Seznam odborné literatury:

Dodá vedoucí práce

Vedoucí: Ing. Ondřej Poláček

Platnost zadání: do konce zimního semestru 2013/2014

  
prof. Ing. Jiří Žára, CSc.  
vedoucí katedry

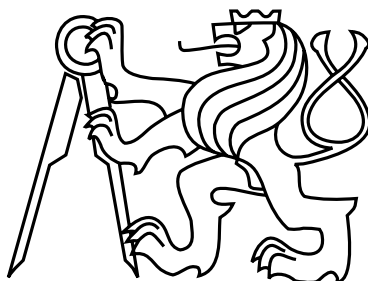


  
prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 15. 10. 2012



České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra počítačové grafiky a interakce



Bakalářská práce

## **Emulace myši a klávesnice pro motoricky postižené uživatele**

*Jan Starý*

Vedoucí práce: Ing. Ondřej Poláček

Studijní program: Softwarové technologie a management, Bakalářský

Obor: Web a multimedia

3. ledna 2013



## Poděkování

Chtěl bych poděkovat všem, kteří mě při tvorbě této práce podporovali a byli mi oporou. Mé díky patří zejména rodině a přítelkyni, mým přátelům, kteří se ochotně zúčastnili testování a samozřejmě panu Ing. Ondřeji Poláčkovi za jeho čas, nápady a nekonečné odhodlání pomoci mi posunout mojí práci zase o krok dál.

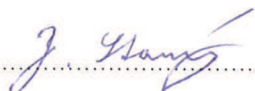


## Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 3. 1. 2013

.....  






# Abstract

This work is focused on creation and subsequent testing of a program emulating mouse and keyboard. It is based upon libraries using input from a webcam and microphone. It describes works specifications and objectives, followed by an actual implementation. The list of used projects comprises of Ing. Ondřej Poláček's VoiceKey and facetracking code from Ing. Ladislav Kunc's Talking Head. These libraries were enriched with computer control functions. Also a new predictive keyboard for Czech language was added.

Testing focuses mainly on usability of the application with regard to the oportunities it brings for disabled people. It also compares the newly created keyboard with Windows 7 on-screen keyboard. Furthermore the advantages that this keyboard brings are discussed.

# Abstrakt

Tato práce se zabývá vytvořením a následným testováním programu emulujícího myš a klávesnici postaveného na základě knihoven využívajících vstupů z webové kamery a mikrofonu. Nejprve je popsána specifikace a cíle, následně pak konkrétní implementace. Práce je založena na projektech VoiceKey Ing. Ondřeje Poláčka a facetrackeru z programu Talking Head Ing. Ladislava Kunce. Tyto projekty byly obohaceny o funkce umožňující ovládání počítače a k celé práci byla navíc vytvořena nová prediktivní klávesnice pro český jazyk.

Následné testování zkoumalo použitelnost aplikace k ovládání počítače tělesně postiženými a porovnání nově vytvořené klávesnice se součástí Windows 7 klávesnice na obrazovce. Dále jsou popisovány výhody, které tato klávesnice přináší.



# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Popis problému, specifikace cíle</b>	<b>3</b>
2.1	Existující metody alternativního ovládní . . . . .	3
2.1.1	Myši . . . . .	3
2.1.2	Klávesnice . . . . .	4
2.1.3	Alternativní polohovací zařízení . . . . .	4
2.1.4	Kompenzační pomůcky . . . . .	5
2.1.5	Softwarové nástroje . . . . .	5
2.2	Cílová skupina uživatelů . . . . .	7
2.3	Problematika emulace myši . . . . .	8
2.4	Problematika emulace klávesnice . . . . .	8
2.5	Propojení a další plán rozvoje . . . . .	8
<b>3</b>	<b>Analýza, návrh řešení</b>	<b>9</b>
3.1	Pohyb kurzoru pomocí kamery . . . . .	9
3.1.1	Výběr facet trackeru . . . . .	9
3.1.2	Převod polohy obličeje na pohyb kurzoru . . . . .	10
3.2	Klikání pomocí neverbálního hlasového vstupu . . . . .	10
3.2.1	Výběr gest . . . . .	11
3.3	Tvorba prediktivní klávesnice . . . . .	11
3.3.1	Predikce . . . . .	11
3.3.2	Rozložení kláves . . . . .	12
3.3.3	Zpětná vazba . . . . .	12
<b>4</b>	<b>Realizace</b>	<b>15</b>
4.1	Získání a využití pozice hlavy . . . . .	15
4.1.1	Pohyb kurzoru . . . . .	16
4.2	Hlasový vstup a gesta . . . . .	17
4.2.1	Emulace funkcí myši . . . . .	17
4.2.2	Úpravy UI . . . . .	18
4.3	Prediktivní klávesnice . . . . .	18
4.3.1	Slovník a trie struktura . . . . .	18
4.3.2	Operace s textem . . . . .	19
4.3.3	Predikce následujícího písmene . . . . .	19

4.3.4	Přidání slov do slovníku . . . . .	20
4.3.5	Rozložení kláves . . . . .	20
<b>5</b>	<b>Testování</b>	<b>23</b>
5.1	Testování s uživateli . . . . .	24
5.1.1	Výběr úkolů . . . . .	24
5.1.2	Testovací podmínky . . . . .	26
5.1.3	Průběh testování . . . . .	26
5.1.4	Výsledky testování . . . . .	32
<b>6</b>	<b>Závěr</b>	<b>37</b>
<b>7</b>	<b>Seznam použitých zkratk</b>	<b>41</b>
<b>8</b>	<b>Instalační a uživatelská příručka</b>	<b>43</b>
<b>9</b>	<b>Obsah přiloženého CD</b>	<b>45</b>

# Seznam obrázků

2.1	Klávesnice MID MEDIUM.[16]	4
2.2	Polohovací zařízení SmartNAV4.[16]	4
3.1	Získání vektoru pohybu kurzoru z pozice obličeje	10
3.2	Neverbální gesta hlasového vstupu.	11
3.3	Zvýraznění pozice obličeje a hraničních os pro směr pohybu kurzoru.	13
3.4	Zpětná vazba tlačítka kopírovat.	13
4.1	Ukázka histogramů snímků obličeje. [2]	15
4.2	Ukázka části trie stromu a obsahu jednoho z uzlů.	18
4.3	Ukázka rozmístění znaků podle jejich pravděpodobnosti.	21
5.1	Průměrný podíl slov doplněných ze slovníku ve 4.úkolu	33
5.2	Ukázka efektivnosti predikce dalšího znaku v úkolu č.4	33
5.3	Průběh psaní jedné věty 2. participantem.	34
5.4	Porovnání časů participantů při psaní v 4. úkole.	34
5.5	Porovnání časů participantů při psaní v 5. úkole.	35
5.6	Průměrný čas psaní na porovnávaných klávesnicích.	35



# Seznam tabulek

4.1	Frekvence bigramů tvořených z O nebo E a písmen A,N,T,S,I,V,L,R. . . . .	21
5.1	Hodnoty WPM u participantů na testovaných klávesnicích. . . . .	36





# Kapitola 1

## Úvod

V dnešní době si už jen málokdo dokáže představit život bez počítačů. Zvláště pro technicky založené lidi je to prakticky nepředstavitelné. Vždyť ho tolik z nás nutně potřebuje k vykonávání své práce, jiní si ji s jeho pomocí ulehčují a ostatní se k němu obrací jako k prostředku, který je spojuje se světem a poskytuje jim informace a nebo vstupuje do jejich životů jako nositel uvolnění a relaxace.

Zkusme si tedy na chvíli představit, jaké to je počítač nemít. V čem by nás to omezovalo? Jak jsem již naznačil, pro některé by to znamenalo nemít práci, druhým by to způsobilo ztrátu obrovské možnosti přístupu k informacím na internetu, pro jiné by to například mohlo znamenat ztrátu určité svobody a nebo možnosti prezentovat sama sebe či kreativně tvořit. Nyní se můžeme v myšlenkách vrátit zpět do světa, ve kterém teď žijeme a můžeme usednout zpět ke svému počítači. Ale co když přijde den, kdy prožijeme velký zvrat. Něco co nám převrátí celý svět vzhůru nohama. Úraz, který nám částečně, nebo úplně znemožní používat horní končetiny. Napadlo vás někdy, že jste to mohli být zrovna vy, nebo vaše děti, kdo se mohl narodit s tělesným handicapem?

Nyní si položíme několik otázek. Jsme schopni handicapovaným pomoci k návratu do plnohodnotného života? Dle mého názoru zní odpověď rozhodně ano. Děláme ovšem vždy maximum pro to, aby i oni mohli využívat naše produkty?

Nejčastěji používanými perifériemi pro ovládání počítače jsou myš a klávesnice. Během let se objevila spousta dalších, ale zdá se, že neotřesitelnou pozici těchto ovladačů ještě dlouho nic neohrozí. Zde tedy vyvstává problém nutnosti fyzického kontaktu uživatele s periférií a tento problém je v mnoha případech pro handicapované uživatele zásadní. Podle informací Českého statistického úřadu [19] žilo v naší zemi v roce 2007 1 015 548 osob se zdravotním postižením, což je 9,87% populace ČR. Nás bude především zajímat číslo 550 407, které zpráva ČSÚ uvádí. Je to počet osob s tělesným postižením. Nemusíme tedy dále rozebírat čísla statistiků, abychom si uvědomili, kolik lidí žije například s postižením horních končetin, nebo poruchou centrální nervové soustavy, která jim znemožňuje používat myš a klávesnici k ovládání počítače.

Abychom pro ně tuto bariéru odstranili, musíme tedy nahradit fyzický kontakt uživatele za jiný prostředek ovládání. Nápadů, jak tento problém vyřešit bylo vymyšleno již mnoho. Počínaje sledováním pozice hlavy nebo zornice oka, přes technologie syntézy a rozpoznávání hlasu, až po zařízení ovládající počítač podle pohybu úst uživatele. Bohužel většina z těchto nástrojů buď nenabízí komplexní řešení, nebo je na pořízení velmi finančně náročná. Uvědomme si, že u těchto uživatelů mohou většinu jejich příjmů tvořit státní příspěvky. Motivací tohoto projektu je tedy dát těmto lidem kvalitní alternativu k již existujícím nástrojům, která by v jednom balíčku poskytovala vše potřebné k nahrazení jak myši, tak klávesnice a zároveň byla dostupná.

### **Struktura práce**

V další kapitole se budu věnovat pojmenování problémů, určení cílů a popisu cílové skupiny uživatelů. Naleznete zde také přehled již existujících programů a zařízení, jejich výhody a nevýhody a informace o jejich dostupnosti. Dále dojde k analýze jednotlivých částí a návrhu jejich řešení. Ve čtvrté kapitole následně popíšu postupy při jeho implementaci. Následně bude popsána forma testování a informace díky němu zjištěné. Vše bude završeno sumarizací poznatků a závěrů z testování s vizí dalšího vylepšení.

## Kapitola 2

# Popis problému, specifikace cíle

V této kapitole bych rád zmínil, co je cílem této práce, jaké jsou existující metody alternativního ovládání počítače a pro které handicapované uživatele je moje aplikace určena. V souvislosti s tím následně uvedu související požadavky a očekávané problémy, které bude potřeba vyřešit. Cílem je vytvořit aplikaci, která by nahradila ovládání počítače myší a zároveň poskytla kvalitní prediktivní klávesnici pro psaní. Programů poskytujících odděleně tyto funkce je nesčetné množství, avšak jen málo z nich nabízí opravdu komplexní řešení za přijatelnou cenu. To je také jedním z důvodů, proč se v následující části budu věnovat přehledu již existujících řešení ovládání počítače pro postižené.

### 2.1 Existující metody alternativního ovládání

Mezi nalezenými způsoby ovládání počítače pro handicapované je mnoho důmyslných hardwarových a softwarových řešení o kterých má cena se zmínit. Podle informací získaných na stránkách občanského sdružení PETIT [16] jsem se zajímal o tyto produkty.

- Speciální myši, trackbally a klávesnice
  - Myš Evoluent, BigTrack, Kensington Orbit Trackball, ...
  - Jumbo XL, BigKeys LX, MID MEDIUM, Maltron One Hand Keyboard, ...
- Alternativní polohovací zařízení
  - Tracker Pro, SmartNAV4, I4Control, Integramouse, ...
- Kompenzační pomůcky
  - Head Pointer

#### 2.1.1 Myši

Do této kategorie patří myši s upravenou ergonomií, se kterými se pohybuje celou paží a ne jen zápěstím. Daleko častěji se však používají trackbally. Uživatel u nich pomocí celé plochy dlaně otáčí velkou kuličku, jejíž pohyby se převádí na pohyby kurzoru. Obě tlačítka mají samozřejmě také dostatečně zvětšené rozměry. Cenově se různé modely pohybují od tisíce do šesti tisíc korun.



Obrázek 2.1: Klávesnice MID MEDIUM.[16]

### 2.1.2 Klávesnice

Co se týká klávesnic mají speciálně upravená tlačítka. Může se jednat o úpravy ve formě dvojnásobně zvětšených kláves, jejich barevného odlišení, nebo změny jejich ergonomického uspořádání. Nevýhodou těchto modelů s 2 - 4cm tlačítky ovšem často bývá, že při snaze zvětšením usnadnit uživateli stisknout správný znak, se výrobce uchýlí k odstranění některých méně používaných znaků. Některé klávesnice obsahují výměnné šablony, pomocí nichž lze individuálně uživateli přiblížit funkci této klávesy, jiné mohou být optimalizovány k ovládání nohou, nebo například bradou. Pro uživatele, kteří mohou používat jen jednu ruku, ale bez jakéhokoli omezení existují i zařízení s rozložením kláves speciálně pro psaní jednou rukou. Všechna tato řešení ovšem připadají v úvahu pouze, pokud je jejich uživatel schopen alespoň částečně používat horních končetin. Zároveň tyto komerční produkty často spadají do cenové kategorie 6 - 15 tisíc korun.

### 2.1.3 Alternativní polohovací zařízení

Mezi ty opravdu alternativní způsoby ovládání lze zařadit zařízení převádějící pohyb hlavy, očí, nebo pohyb úst na pohyb kurzoru. Sem patří například zařízení SmartNAV4, Tracker Pro, I4Control a Integramouse. První dvě využívají principu vysílání infračerveného paprsku a sledování jeho zpětného odrazu od reflexního bodu na čele uživatele. Nevýhodou reflexních bodů prý je, že špatně drží na čele uživatele. Tato zařízení je navíc vždy potřeba kombinovat se softwarem, který následně řeší stisknutí tlačítek.



Obrázek 2.2: Polohovací zařízení SmartNAV4.[16]

Na rozdíl od toho I4Control[1] přichází s malou kamerou uchycenou na obroučkách brýlí. Tato kamera sleduje pohyb zornice oka a získané informace používá k vyhodnocení pohybu kurzoru. Dokáže také rozlišit přirozené mrkání oka od dlouhého mrknutí, které tím pádem může využívat jako povel pro kliknutí. Potěšující je i to, že se jedná o český výrobek. Nevýhodou může být, že každý uživatel potřebuje jinak velké obroučky brýlí.

Posledním zařízením je Integramouse, což je speciální ovladač řízený drobnými pohyby úst. Princip jeho fungování je velmi podobný joysticku, který je tvarem upraven tak, aby se dal vložit do úst. Při změně uživatele je potřeba dbát na hygienu a vyměnit část ovladače, který přichází do kontaktu s ústy.

Bohužel jsou tyto produkty často náročné na vývoj a tak se jejich cena pohybuje v desítkách tisíc korun.

#### 2.1.4 Kompenzační pomůcky

Poslední z možností je použití nástroje spadající do kategorie kompenzačních pomůcek. Mohou to být například speciálně vytvořená ukazovátka uchycená na hlavě, pomocí nichž je člověk schopen stisknout klávesy na standardní klávesnici, nebo jsou k dostání různé formy krytů a podložek ke klávesnici zabraňující nechtěnému stisku tlačítka. Velmi dobře je mohou používat například lidé s problémy jemné motoriky.

#### 2.1.5 Softwarové nástroje

Do této skupiny patří obrovské množství již existujících programů a to počínaje těmi, které jen napomáhají uživateli lépe využívat standardní metody ovládání počítače až po jiné, které tyto metody kompletně nahrazují. Patří sem například i funkce usnadnění přístupu obsažená v operačním systému Windows.

#### Nastavení v OS Windows

Microsoft nám přímo v operačním systému, tedy bez nutnosti jakékoli instalace dodatečného softwaru přináší několik zajímavých funkcí. V nastavení myši lze například nastavit povolenou prodlevu mezi jednotlivými stisky v rámci dvojkliku. Co se týče možností, které mohou handicapovaní využít je velmi zajímavé hlavně Centrum usnadnění přístupu.

Tou neznámější pomůckou je asi klávesnice na obrazovce. Tato klávesnice byla po dlouhou dobu opomíjena a bohužel se jí nedostávalo aktualizací. S příchodem posledních verzí tohoto systému se naštěstí vývoj těchto komponent posunul opět kupředu. Více se o ní zmíním v další části této práce. Abych se vrátil k tomu, co dalšího toto centrum nabízí. Jedná se hlavně o možnosti nahrazující funkce myši a klávesnice, jako například pohyb kurzoru na numerické klávesnici, o vizuální úpravy, jako je změna velikosti kurzoru nebo o program Narrator schopný předčítat text na obrazovce. A samozřejmě je zde i lupa a funkce Jedním prstem, která umožňuje právě jediným prstem stisknout i složité klávesové zkratky. Pokud máte zájem o více informací naleznete je v příslušné sekci webu OS Windows. [12]

## Softwarové klávesnice

Jak vlastně vybrat správnou softwarovou klávesnici? Proč nám nestačí, že některé z programů nahrazují pouze funkce obyčejné klávesnice? Na obyčejné klávesnici můžeme psát znaky v rychlém sledu za sebou až všemi deseti prsty. Na těch softwarových však musíme psát kurzorem vždy jen po jednotlivém znaku. Velmi také záleží na rozložení znaků na klávesnici, které daný program využívá. Čím jsou tlačítka menší, tím menší dráhu je nutné při přejetí kurzorem urazit. Protože obyčejné softwarové klávesnice přestávají dostatečně požadavkům uživatelů, začínají se čím dál tím více objevovat vylepšené takzvaně prediktivní klávesnice, které se snaží různými metodami zrychlit a zefektivnit psaní.

Dle mého názoru k rozvoji prediktivních klávesnic velmi přispěly mobilní telefony a jejich snaha efektivně nabídnout psaní na malém prostoru ať už klávesnice nebo později displeje telefonu. Tyto klávesnice se snaží za pomoci pravděpodobnostních jevů v jazyce z rozepsaného textu a integrovaného slovníku nabídnout nebo případně doplnit právě to slovo, které se uživatel snaží napsat. Že je do velké míry rychlost psaní ovlivněna rozložením znaků na klávesnici se můžeme přesvědčit z práce pánů Zhaie, Huntera a Smitha.[18] Ti ve své práci rozebírají počet slov napsaných za minutu (WPM) u klávesnic s různými layouty a snaží se najít rozložení optimální.

Mezi tyto softwarové klávesnice jako jsou Click N-Type Keyboard, Free Virtual Keyboard a Microsoft on-screen keyboard patří i jedna s úplně odlišným přístupem k psaní. Jedná se o aplikaci nazvanou Dasher.[10] Slovo je psáno vlastně tak, že každý znak má svojí barevně ohraničenou oblast, jejíž velikost odpovídá jeho pravděpodobnosti v textu. Když zaměříme kurzor na písmeno, daná oblast se nám přiblíží a nabídne nám další možná písmena ve vlastních příslušně velkých oblastech. Lze tedy psát celé věty pouhým pohybem kurzoru. A v neposlední řadě můžeme narazit na klávesnice T9, které jsou ovšem spíše doménou mobilních telefonů. Díky predikci je na nich možné psát slova jen za pomoci devíti tlačítek.

## Programy pro emulaci klikání

Existuje opět spousta možností, jak nahradit tlačítka na myši. Některé programy například mapují stisknutí tlačítek myši na klávesnici, což je samozřejmě problém ve chvíli, kdy uživatel není schopný používat ani klávesnici. Pak přichází na řadu například program DwellClick. Tento program funguje na principu jednoho právě zvoleného režimu, kterým může být stisknutí levého tlačítka, pravého tlačítka, dvojklik a nebo drag and drop. Při ponechání kurzoru v klidu se po uplynutí předem nastaveného času vyvolá daná akce. Cenový rozptyl těchto aplikací je od stovek do tisíců korun. Navíc u nich dochází k tzv. Midas touch problému [9], tedy k nechtěnému klikání z důvodu neaktivity kurzoru. Proto je u těchto programů nutností vytvářet místa pro "zaparkování kurzoru".

## Programy pro rozpoznání a syntézu řeči

Jedná se o velmi složitou a obsáhlou oblast a prostor pro vědecké týmy k neustálému zlepšování. Lidská řeč, množství existujících jazyků a jejich unikátní prvky. To vše je zdrojem spousty problémů k řešení. Když vezmeme v potaz, že každý člověk má jiný hlas, lidé mají přízvuk podle jejich rodné oblasti a jazyk, kterým komunikují s programem, nemusí být jejich mateřský, je až z podivem, jak dobře některé z moderních systémů fungují.

Mezi těmi rozpoznávajícími řeč existují takové, které jsou založeny na předdefinovaných datech od výrobce, a váš hlasový vstup je s nimi porovnáván. Další možností je nahrání vlastních hlasových příkazů a jejich následné porovnávání se zvukovým vstupem. Dále lze pak rozlišovat, jestli je počítač ovládán pomocí slov, nebo pomocí neverbálních zvuků a hluků. [13]

Abych zmínil alespoň některé konkrétní aplikace, jedná se například o program MyDictate [3] umožňující napsat jakýkoli text za předpokladu, že ho diktujete postupně s pauzami po jednotlivých slovech. Dále jsou zde programy jako JetVoice [4], u kterých si nejprve sami vytvoříte svoji databázi hlasových povelů a pak díky nim můžete ovládat široký záběr aplikací a funkcí. A nebo máte šanci využít aplikace typu MyVoice [3], která je díky svým povelům schopná nahradit funkce myši i klávesnice. K předčítání lze použít například Microsoft Narrator, o kterém již padla zmínka v textu o centru usnadnění přístupu.

### Programy reagující na neřečový hlasový vstup

Anglický výraz humming označuje hlasové ovládání založené právě na neverbálních zvucích. Jde tedy většinou o ovládání pomocí tónu [15]. Dalšími možnostmi, jak ovlivňovat tento vstup jsou změna délky, hlasitosti a nebo barvy tónu. Tyto programy fungují na základě vytvoření gest, které popisují statický nebo dynamický stav těchto proměnných a přiřazují k nim určitou akci.

Tato metoda byla například využita v projektu Humsher, který se zabývá vytvořením prediktivní klávesnice ovládané právě houkáním [14]. Další existující klávesnici využívající tento design je pak QANTI. Jednoznačnou výhodou neverbálního vstupu je odstranění jazykové bariery a rychlost ovládání v případě jednoduchých gest. Naopak je nevýhodou, pokud je gest velké množství a některá jsou tím pádem složitá a špatně zapamatovatelná.

## 2.2 Cílová skupina uživatelů

Moje snaha vede k vytvoření produktu, který umožní uživatelům s handicapem každodenní běžnou práci na počítači. Aby byl můj cíl a jednotlivé jeho pod-úkoly správně definován, je potřeba se nejprve zaměřit na cílovou skupinu uživatelů a pochopit jejich potřeby a problémy. Teprv s těmito znalostmi lze následně navrhnout to správné řešení, které jim bude při používání vyhovovat.

Existuje velké množství handicapů, které se dá rozdělit do určitých základních skupin. I přes toto rozdělení má každý uživatel s postižením specifické problémy a potřeby. Proto je velmi těžké, ne-li nemožné vytvořit jedno vše pojímající řešení. Z toho důvodu bych se chtěl zaměřit na vytvoření produktu hlavně pro následující skupiny uživatelů.

- Monoparetici a kvadraparetici [20], tedy osoby s částečnou ztrátou schopnosti pohybu jedné, nebo všech čtyřech končetin.
- Lidé s roztroušenou sklerózou.
- Uživatelé s deformací nebo po amputaci jedné či obou horních končetin.

Produkt by mohl být přínosný i pro další skupiny jako jsou lidé s kontrakturou, tedy zkrácením a částečnou přestavbou svalů horních končetin a nebo svalovou atrofií.

## 2.3 Problematika emulace myši

Naši uživatelé nebudou pravděpodobně schopni v efektivní míře využívat klasické polohovací zařízení. Z tohoto důvodu vzniká potřeba šetrně a podle možností uživatelů získávat nějakou formou informace o pohybu části jejich těla. Z těchto údajů lze následně sledovat změnu pozice, kterou můžeme vyhodnocovat jako povel k změně polohy kurzoru myši. Pro tento účel jsem obdržel kód pro rozpoznání obličeje postavený na základě OpenCV 2.1 [6], který pracuje se vstupem z webové kamery. OpenCV je knihovna obsahující funkce pro počítačové vidění. Na výstupu rozpoznání obličeje získáme souřadnice o bodu, který byl označen za střed polohy hlavy. Je tedy potřeba stanovit správný postup využívání těchto informací ke změně pozice kurzoru.

Podobně tomu bude i u emulace tlačítek. Zde byl součástí zadání kód pro rozpoznání zvukového vstupu na základě výšky jeho MIDI tónu. Musíme tedy stanovit určitá jednoduchá gesta, která se následně namapují k příslušné akci stisknout tlačítko. Jelikož chceme co nejvíce eliminovat případnou nutnost zásahu další osoby do ovládání, bude vhodné namapovat na jedno z gest kalibraci systému pro pohyb kurzoru. Všechna gesta bychom měli zobrazit, aby uživatel netápal, které má v jaké situaci použít. Zároveň bych chtěl zachovat zpětnou vazbu při rozpoznání tónu, která je v zadání implementována

## 2.4 Problematika emulace klávesnice

V rámci práce bylo potřeba vytvořit i softwarovou klávesnici, která by eliminovala fyzický kontakt při psaní textu. Požadovat od ní budeme pohodlné a rozumně rychlé psaní, nápovědu slov ze slovníku a dostatečně velká tlačítka, aby nedocházelo k překlepům. Základní předpoklad také byl, že celá klávesnice bude ovládána pouze pomocí kurzoru. Všechny její funkce by měly být snadno a rychle dostupné. Bude potřeba zvolit, případně vymyslet vhodné rozložení kláves tak, aby toto rozložení co nejvíce podpořilo psaní v českém jazyce. Další problém, jehož řešením se budu zabývat je přenos textu napsaného na klávesnici do příslušného okna pro které je tvořen. K zmíněnému slovníku je potřeba navrhnout, jak budou jednotlivá slova reprezentována v paměti a jak se v nich bude vyhledávat a vkládat mezi ně slova nová.

## 2.5 Propojení a další plán rozvoje

Všechny dílčí prvky této práce bude potřeba propojit tak, aby bylo spuštění výsledného produktu jednoduché a aby vše spolupracovalo. Celková funkčnost práce bude ověřena v testu s uživateli se simulovaným postižením. Na základě tohoto testu zjistíme, jaké jsou výhody a nevýhody návrhu a kterým směrem by se měl vývoj aplikace dále ubírat.



## Kapitola 3

# Analýza, návrh řešení

Nyní rozeberu aplikaci na jednotlivé části a postupně se budu věnovat jejich návrhu a porovnání jednotlivých možných řešení. Konkrétní implementaci pak popíši až v další kapitole práce.

Nejprve se podíváme na rozpoznání obličeje a s ním spojené ovládání kurzoru. Následně rozpoznání houkání, tedy neverbálního zvukového vstupu uživatele a na závěr si nechám rozbor vzniku prediktivní klávesnice vytvořené pro tuto práci.

### 3.1 Pohyb kurzoru pomocí kamery

Jak jsem se již zmínil, mým úkolem je získat od uživatele informace, že chce pohnout kurzorem. Jako prostředník v komunikaci uživatele s počítačem, respektive s tímto programem, je potřeba vhodné vstupní zařízení. Prostředkem získání těchto informací byla hned v samém začátku práce zvolena webová kamera. Velkou výhodou této volby je, že dnes je toto zařízení velmi dostupné a je součástí prakticky všech dnes prodávaných notebooků. Navíc je to zařízení, které od uživatele v rámci našeho použití nebude vyžadovat žádný fyzický kontakt. Jsem plně přesvědčen, že je tato metoda schopná splnit potřebné nároky, jak na přesnost, tak i na rychlost pohybu kurzoru a zároveň věřím, že bude tento způsob ovládání velmi intuitivní.

#### 3.1.1 Výběr facetrackeru

Nejprve bylo počítáno s využitím OpenGazeru [7], open-source programu pro trackování pozice hlavy. Tato volba se ovšem časem ukázala jako neoptimální a finálním rozhodnutím pro použití jiného trackovacího systému bylo koncipovat celý projekt pro uživatele OS Windows. OpenGazer je totiž bohužel čistě záležitostí Linuxu. Najít jiný trackovací program vhodný pro rozšíření a současně zdarma dostupný ovšem nebylo jednoduché.

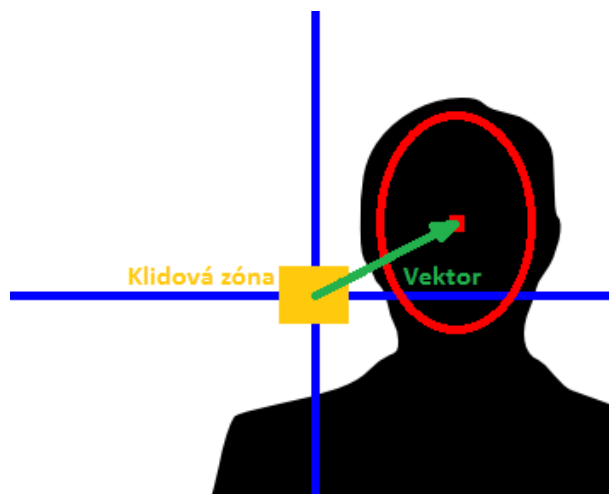
Nakonec jsem dostal jako základ pro sledování hlavy projekt Face Detection Service psaný v jazyku C++ a používající již zmíněnou knihovnu OpenCV 2.1. Tento projekt se v nekonečném cyklu pokouší sledovat v záznamu z kamery obličej a vypisuje jeho polohu, tedy střed elipsy, v které byl obličej nalezen, do konzole. V OpenCV je celý proces trackování hlavy založen na algoritmu Camshift [2]. Implementaci facetrackeru nebylo potřeba v rámci

této práce nijak upravovat. Byla nad ním tedy jen vytvořena nadstavba využívající informace o pozici obličeje.

### 3.1.2 Převod polohy obličeje na pohyb kurzoru

Face Detection Service tedy poskytuje na výstupu informace o pozici středu obličeje na osách X a Y. Navíc udává i úhel natočení obličeje ke kameře, u kterého bohužel nelze rozlišit mezi osami a proto jsem ho nijak nevyužil.

Způsobů, jak řešit pohyb kurzoru, je několik. V průběhu této práce jsem postupně vyzkoušel tři. První metoda bylo v podstatě čisté zobrazení relativní pozice obličeje v rámci bodů získávaných z kamery do prostoru bodů na obrazovce. Tuto metodu jsem záhy zavrhl z důvodu, že se u ní trhavým pohybem kurzoru projevovaly nepřesnosti v sledování obličeje. Druhá metoda dělila obraz z kamery na čtyři kvadranty. Pokud uživatel vyklonil svoji hlavu do jednoho z kvadrantů více než byl záběr klidové zóny okolo středu obrazovky, začal se kurzor daným směrem pohybovat. Velikost pohybu byla závislá na míře vyklonění hlavy. Tento koncept byl nakonec upraven a to z důvodu umožnit uživateli mít klidovou zónu pro kurzor přesně tam, kde je při ovládání počítače zvyklý sedět. Pozice obličeje totiž často nemusí být ve středu před kamerou a to hlavně vertikálně.



Obrázek 3.1: Získání vektoru pohybu kurzoru z pozice obličeje

## 3.2 Klikání pomocí neverbálního hlasového vstupu

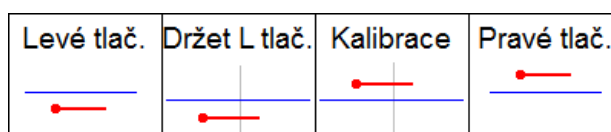
Pro nahrazení klasického stisknutí tlačítka myši byla vybrána metoda používající neverbální hlasový vstup. Její výhodou je jednoduchost na ovládání a nenáročnost na zapamatování si několika gest. Podle zjištěných informací [13] by měla být tato metoda v dlouhodobém používání výhodnější, než automatické rozpoznávání hlasu a zároveň nám při používání neklade žádné jazykové bariéry. I vzhledem k tomu, že nám budou stačit přibližně čtyři gesta, je pro naše použití ideální. Myslím, že toto řešení bude ve spojení s ovládáním kurzoru po-

hybem hlavy výhodnější, než v tomto případě také často používaný dwell clicking založený na ponechání myši na tlačítku po určitý čas.

### 3.2.1 Výběr gest

Při výběru gest jsem přemýšlel nad tím, jak uživateli umožnit použití co nejširšího spektra ovládacích funkcí, při zachování malého počtu gest. Dlouze jsem se tedy rozhodoval, které funkce mají prioritu a které se dají snadno nahradit nebo obejít. Nakonec jsem dospěl k názoru, že kromě obvyčejného kliku levým a pravým tlačítkem myši by bylo dobré implementovat gesto pro drag and drop, respektive pro držení levého tlačítka. Ať už chce totiž uživatel pohybovat okny, přesouvat soubory, scrollovat obsah stránky nebo označovat větší počet položek ušetří mu tato funkce spoustu času i námahy.

Vytvořit čtyři různá gesta pomocí tónu už bylo jednoduché. Za pomoci nastavitelné mezní hodnoty můžeme rozhodnout, jestli se jedná o vysoký a nebo nízký tón. Tato hodnota musí být nastavitelná tak, aby vyhovovala hlasové dispozici jedince. Další dvě gesta získáme jednoduchým porovnáním, jestli je tento tón dlouhý a nebo naopak krátký. Protože stisknutí levého a pravého tlačítka budou dvě nejčastější operace, namapoval jsem je na krátký nízký respektive vysoký tón. Dále už bylo logické pro držení levého tlačítka zvolit stejný tón jako pro jeho stisknutí, jen s delším trváním. Jako poslední gesto zbyl dlouhý vysoký tón, který byl nakonec přiřazen ke kalibraci systému rozpoznávání obličeje a pohybu kurzoru. Touto kalibrací je uživatel kdykoli schopen změnit polohu klidové zóny a os určujících směr pohybu.



Obrázek 3.2: Neverbální gesta hlasového vstupu.

## 3.3 Tvorba prediktivní klávesnice

Jedním z požadavků na tuto práci bylo vytvoření prediktivní klávesnice, která bude uzpůsobena alternativnímu ovládání pomocí kamery. Při návrhu jsem musel mít od začátku na paměti, že by bylo vhodné vytvořit klávesnici s většími tlačítky, aby tak nedocházelo k překlepům. Zároveň jsem se rozhodl, že jakožto z projektu snadno vyjmutelnou součást vytvořím klávesnici v Javě. A to proto, aby byla multiplatformní a mohla být případně využita i samostatně. Vzhledem k velikosti tlačítek by ji mohli využít například lidé se zrakovými problémy.

### 3.3.1 Predikce

Prediktivní klávesnice pracují na principu rozboru právě psaného textu, snaží se dopředu odhadnout jeho pokračování a nabídnout vám například doplnění slova podle slovníku. V návrhu své klávesnice jsem se také rozhodl využít slovníku, který je při běhu uchovávan v paměti ve stromové struktuře. Původním nápadem k sestavení nápovědy bylo využít Ngramy[5]

vytvořené scanováním knih společností Google. Bohužel jsou tyto Ngramy zatím dostupné pouze pro cizí jazyky. Do budoucna to ale určitě bude zajímavou myšlenkou ke zpracování.

Pro tvorbu této klávesnice jsem nakonec využil soubor obsahující základní slovník českých slov a částečně i informace o frekvenci jejich výskytu. Podle této frekvence jsou slova seřazena v nabídce. Z důvodu, že se v dnešní době hodně slov přejímá a vzniká mnoho nových, bylo rozhodnuto, že musí mít slovník schopnost ta nová ukládat. Při každém doplnění slova z nabídky slovníku jsou také automaticky zobrazeny i možnosti ukončujících znaků věty a to i s možností přidružené mezery za nimi.

Dalším využitým prvkem predikce je nápoředa dalšího nejpravděpodobnějšího znaku. Má formu nápadného barevného zvýraznění až tří nejčastěji následujících písmen po posledním napsaném znaku. Všechny informace o této pravděpodobnosti jsou opět získávány z použitého slovníku.

### 3.3.2 Rozložení kláves

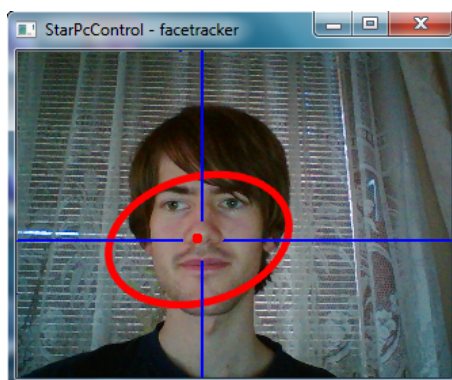
U každého rozložení, neboli layoutu znaků lze najít jeho výhody i nevýhody. U virtuálních klávesnic se objevuje například ten problém, že jimi používaný layout je prostou kopií těch používaných na klávesnicích hardwarových. Tyto layouty byly ovšem navrženy pro psaní deseti prsty tak, aby například u QWERTY ze základní pozice s ukazováčky na písmenech F a J byla všechna písmena rychle dostupná. S tím také souvisí větší šířka než výška celé klávesnice. Pro virtuální klávesnice by ovšem mělo být lepší, když je vzdálenost všech znaků od středu co nejmenší, tedy layout do čtverce nebo kruhu.

O různých rozloženích kláves se můžete více dočíst v [18]. Bohužel zmiňované layouty jsou použitelné pro anglickou klávesnici a my používáme češtinu. Autoři zde také zmiňují zajímavý fakt, že klasická QWERTY klávesnice má znaky tvořící ty bigramy, které se v jazyce vyskytují s velkou pravděpodobností, na opačných stranách klávesnice. To nám vůbec nevyhovuje, protože máme jen jeden kurzor a ne dva. Z tabulky frekvence použití jednotlivých znaků a bigramů v češtině [8] jsem navíc nabil dojmu, že na QWERTZ layoutu je umístění všech znaků obsahujících háčky a čárky vzhledem k jejich pravděpodobnostem nešťastné. Vždyť například í je umístěno až na kraj na klávesu s číslovkou 9 a to i přes to, že je nejpravděpodobnějším dlouhým písmenem.

Proto jsem se rozhodl vytvořit svůj vlastní layout znaků a to právě na základě frekvencí výskytu v jazyce. Mimo to bude můj layout částečně brát v potaz i frekvence bigramů. Budu se tedy snažit o to, aby výsledné rozložení znaků co nejvíce zkrátilo dráhy, které při přejíždění mezi písmeny musí kurzor urazit.

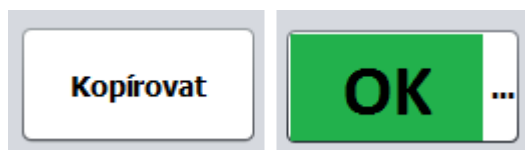
### 3.3.3 Zpětná vazba

U aplikací je samozřejmě velmi důležité myslet i na zpětnou vazbu. Rád bych tedy zmínil, jaké prvky zpětné vazby jsem se rozhodl použít. Od počátku návrhu jsem chtěl ponechat viditelné okno s obrazem z kamery, na němž je elipsou ohraničený obličej. Podle něj může uživatel kdykoli zjistit, jestli sedí v záběru a zda je jeho obličej facetrackerem správně rozeznáván. Později bylo toto zobrazení doplněno o osy určující směr pohybu kurzoru a také barevný bod znázorňující střed obličeje.



Obrázek 3.3: Zvýraznění pozice obličeje a hraničních os pro směr pohybu kurzoru.

Dále jsem z původní aplikace pro neverbální hlasový vstup zachoval zvýrazňování aktuálně rozpoznávaného gesta jeho vybarvením šedou barvou. V tomto trendu jsem se snažil pokračovat i při tvorbě klávesnice. U ní se při najetí kurzorem barevně zvýrazňují nabízená slova ze slovníku, díky čemuž má uživatel přesně přehled, které bude po stisknutí tlačítka doplněno. Na konci napsaného textu se také zobrazuje blikající kurzor, aby bylo možné poznat, kam se bude nový text vkládat a zda uživatel napsal za koncem slova mezeru či ne. Po zkušenosti z pilotního testu jsem pak přidal zabarvení se zprávu OK po stisknutí tlačítka kopírovat. Několikrát jsem se totiž dostal do situace, kdy jsem si nebyl jist, jestli jsem dané tlačítko stiskl nebo ne.



Obrázek 3.4: Zpětná vazba tlačítka kopírovat.



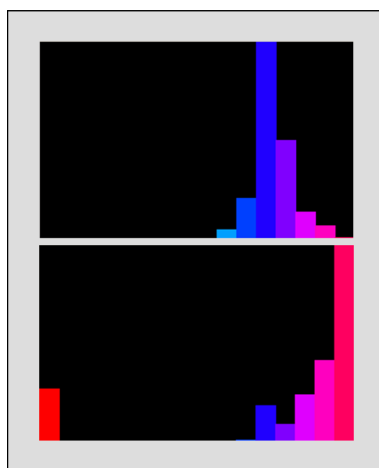
# Kapitola 4

## Realizace

### 4.1 Získání a využití pozice hlavy

Jak tedy funguje sledování pozice hlavy ve Face Detection Service? Stačí nahlédnout do jeho kódu nebo si přečíst něco o trackování s OpenCV [2]. Nejprve inicializujeme instanci FaceTrackeru a na něm voláme ve smyčce metodu `DetectFace()`. Tato metoda se snaží najít v jednotlivých snímcích videa stopy po objektu, v našem případě lidském obličeji, a to podle instrukcí ze souboru `haarcascade_frontalface_default.xml`. Pokud obličej nalezne, dojde k ukončení smyčky.

Nyní začíná vlastní trackování. Ze snímku obličeje je vytvořen histogram, který nám říká, jaké barevné odstíny a v jakém množství se na snímku vyskytují. Následně je vytvořena další nekonečná smyčka a v ní na FaceTrackeru volána metoda `trackOne()`. Tato metoda vždy požádá o další snímek videa a u každého jeho pixelu zjistí odstín barvy. Poté je z histogramu obličeje a odstínu pixelu určena pravděpodobnost s jakou se jedná o jeden z pixelů obličeje. Z informací o pravděpodobnosti výskytu obličeje ve všech bodech snímku už jen následně Camshift algoritmus určí nový střed obličeje.



Obrázek 4.1: Ukázka histogramů snímků obličeje. [2]

### 4.1.1 Pohyb kurzoru

Výsledné řešení pohybu kurzorem vypadá tak, že se nejprve provede inicializace nastavení jeho rychlosti a velikosti klidové zóny ze souboru settings.ini. Následně se provede kalibrace, kdy se nastaví počátek soustavy souřadnic do bodu s nalezeným obličejem a ve stejnou chvíli je kurzor přesunut do středu obrazovky.

```
//set origin
origin_x = x;
origin_y = y;
```

Nyní při každé přichozí informaci o pozici obličeje vypočteme vzdálenost X a Y souřadnice tohoto bodu vůči počátku. Získáme tak vektor, kterým chceme pohybovat s kurzorem. Pokud je vektor delší než poloměr klidové zóny, upravíme jeho velikost pomocí konstanty ovlivňující rychlost pohybu kurzoru a následně přičteme jeho příslušné složky k aktuální pozici kurzoru myši. Na tuto novou pozici pak pomocí WinAPI [11] funkce `SendInput()` přemístíme kurzor. V případě, kdy je vektor kratší než poloměr klidové zóny k pohybu kurzoru nedochází. Tato vlastnost je implementována z toho důvodu, aby bylo pro uživatele snazší udržet kurzor na jednom místě.

```
//Count new cursor position
cursor_x = cursor_x + (cursorSpeed * distance_x);
cursor_y = cursor_y - (cursorSpeed * distance_y);

//WinAPI move cursor
INPUT input;
ZeroMemory(&input, sizeof(input));
input.type = INPUT_MOUSE;
input.mi.mouseData=0;
input.mi.dx = cursor_x*(65535.0f/(width-1));
input.mi.dy = cursor_y*(65535.0f/(height-1));
input.mi.dwFlags = MOUSEEVENTF_ABSOLUTE | MOUSEEVENTF_MOVE;
SendInput(1,&input,sizeof(input));
```

V neposlední řadě jsem vytvořil metodu `resetCamCentrePoint()`, jejíž spuštění jsem navázal na stisknutí klávesy F12. Důvodem její existence je potřeba znovu spustit kalibraci a umožnit tak pohodlné užívání programu. Jak bude později zmíněno tuto kalibraci je možno spustit pomocí zvukového gesta tak, aby to uživatelé s postižením zvládli sami bez pomoci.



## 4.2 Hlasový vstup a gesta

Jako základ pro tuto část jsem obdržel projekt VoiceKey psaný v jazyce C++. Jeho funkcí je čtení a zpracovávání informací z mikrofonního vstupu. Zpracování je založené na samplování vstupního signálu. Ze získaných vzorků je u každého možné zjistit výšku tónu. Následně přichází na řadu algoritmus, který porovnává sled tónů v po sobě získaných vzorcích s popisem jednotlivých gest. Gesta samotná jsou popsána pomocí formálního jazyka, o kterém se zde více dozvědět v [13].

Pro ukázkou uvádím, jakou podobu ve formálním jazyce dostávají gesta, která jsou v této práci použita.

```
"p1 [p1.m < $THRESHOLD]"
" ( p2 *100;500 s <left_click>"
" | p3 *500; s <left_hold>) |"
"p4 [p4.m >= $THRESHOLD]"
" ( p5 *100;500 s <right_click>"
" | p6 *500; s <calibration>);"
```

Tedy pokud je splněna podmínka [p1.m < \$THRESHOLD], pak může nastat právě jedna z dvou akcí left\_click nebo left\_hold. Treshold je předem stanovená mezní hodnota pro rozpoznání vysokého tónu od nízkého. Tato hodnota je reprezentována pomocí MIDI tónu. p2 \*100;500 s <left\_click> nám říká, že pokud je tento tón pod mezní hodnotou a zároveň 100-500ms dlouhý, je označen za krátký a je vyvoláno stlačení levého tlačítka. Naopak zápis p3 \*500; s <left\_hold> označuje, že je-li tento tón delší než 500ms, pak ve chvíli kdy přijde signál s, znamenající ticho z anglického slova silence, dojde k stisknutí a držení levého tlačítka. V druhém případě, kdy je [p4.m >= \$THRESHOLD] musíme opět zjistit, zda se jedná o krátký tón p5 \*100;500 s <right\_click> a stisknout pravé tlačítko myši, nebo tón delší než 500ms p6 \*500; s <calibration> a spustit kalibraci.

### 4.2.1 Emulace funkcí myši

Dále jsem naimplementoval samotné funkce emulující myš, které jsou spouštěné pomocí zmíněných gest. K jejich implementaci bylo opět potřeba studovat další podrobnosti o funkcích WinAPI na stránkách MSDN [11].

Byla opět použita funkce SendInput(), pomocí které jsou nahrazeny funkce levého a pravého tlačítka. Použití v těchto případech se liší pouze uvnitř struktury MOUSEINPUT mi;. V ní jsou použita rozdílná označení dwFlags, tedy informace o kterou událost se jedná. Například pro stisk levého tlačítka je to Input.mi.dwFlags = MOUSEEVENTF\_LEFTDOWN;.

U metody CalibrateFaceDet() bylo nutné použít další z ještě nezmiňovaných funkcí. V první řadě jsem použitím FindWindow() získal referenci na okno aplikace. Za pomoci další funkce a to konkrétně SetActiveWindow() jsem toto okno nastavil jako aktivní, abych do něj mohl následovně posílat stisky kláves. Jako volná klávesa k namapování kalibrace byla již v programu na neverbální hlasový vstup zvolena funkční klávesa F12. Pomocí gesta na kalibraci chci tedy stisk této klávesy poslat do okna rozpoznávání obličeje. K tomu mi pomůže funkce keybd\_event(), které jako parametr předám virtuální kód tlačítka, tedy VK\_F12, (BYTE)VK\_F12 a zároveň pokud se jedná o uvolnění tlačítka i označení události KEYEVENTF\_KEYUP.

## 4.2.2 Úpravy UI

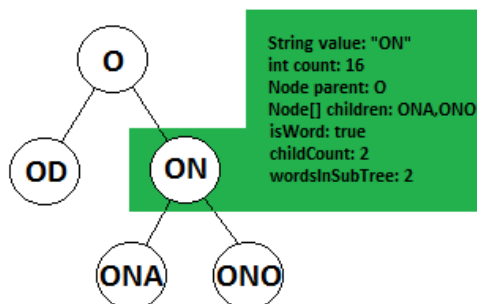
Nakonec bylo potřeba provést jemné úpravy v uživatelském rozhraní tak, aby uživatel viděl správné popisky jednotlivých funkcí gest. Samozřejmě je nutné mít tyto popisky v českém jazyce a vyřešit jak je pomocí zkratk nazvat. Byla též potřeba vytvořit nové obrázky pro pochopení kterými tóny se které gesto používá. Při tvorbě těchto obrázků jsem se inspiroval původním konceptem využitým již v prvotně obdrženém projektu. Dva obrázky z něj jsem zachoval a druhé dva vytvořil ve stejném duchu. Snažil jsem se dodržet jednotný vzhled a snadné porozumění těmto piktogramům.

## 4.3 Prediktivní klávesnice

### 4.3.1 Slovník a trie struktura

Slovník mé prediktivní klávesnice je postaven na ukládání slov do datové struktury s názvem trie. Jedná se o strom jehož uzly obsahují textové řetězce. Pro každý rodičovský uzel zároveň platí, že je jeho řetězec prefixem řetězců všech jeho potomků. Tuto strukturu jsem celou od základu naimplementoval i s následující metodou `add(String word, int count)`, která přidá nové slovo a počet jeho použití v proměnné `count`. Zároveň se postará o inkrementaci proměnné `wordsInSubTree`, která poskytuje počet slov v daném podstromu. Další dvě metody slouží k vyhledávání slov. `getNodeNamed(String input)` dostane jako parametr řetězec a hledá v trie uzel se stejným jménem. Na základě nalezení tohoto uzlu metoda `getWords(String input)` zajistí rekurzivní prohledání podstromu a vrátí list všech slov pro nápovědu.

Uzly této struktury jsou naimplementovány tak, že každý z nich obsahuje uložený řetězec, tedy slovo nebo prefix. Dále pak udržují odkazy na rodičovský uzel, pole potomků a boolean sloužící k rozpoznání prefixu od plnohodnotného slova s významem. Obsahuje i `int wordsInSubTree`, k jehož použití se ještě vrátím u predikce dalšího znaku.



Obrázek 4.2: Ukázka části trie stromu a obsahu jednoho z uzlů.

### 4.3.2 Operace s textem

Text napsaný na klávesnici bude v paměti reprezentován jako textový řetězec `String text`. Základní funkci psaní obstarávají action listenery navázané na jednotlivá tlačítka. Při stisku tlačítka je pak volána metoda `pushKey(int key)`, které je předána ASCII hodnota malého či velkého písmene v závislosti na hodnotě boolean `shift`.

Z dalších operací s textem bych zmínil metodu `getLastWordFragment(String s)`, jejíž úkolem je získat poslední napsané nebo rozepsané slovo. Je využívána při mazání slov nebo jejich doplňování ze slovníku. Jelikož klávesnice podporuje napsání většího počtu řádků, musíme nejprve pomocí `split("\n");` získat poslední řádek. Nyní už stačí jen v tomto řetězci, pokud je zde obsažen, najít poslední oddělovací znak. Tedy největší číslo z množiny poslední pozice výskytu tečky, čárky, mezery, vykřičníku nebo otazníku. Na základě znalosti už jen stačí vybrat dílčí řetězec, který za touto pozicí začíná. Podobně funguje i metoda `delWord(String s)`, která ovšem zachová pouze tu část řetězce před pozicí posledního oddělovacího znaku včetně.

Jako poslední zmíním metodu `concatDictWord(String text, Object selectedValue)` pro doplnění slova vybraného ze slovníku. Jako parametry dostává celý aktuálně napsaný text a objekt, na který uživatel klikl v nabídce slov. Rozepsaná část slova je smazána pomocí `delWord()` a z objektu je získán potřebný řetězec se slovem. Pokud není prázdný je připojen na konec textu.

Zároveň uvnitř kódu používám ještě dvě podmínky. První vznikla z nutnosti označit doplnění mezery v nabídce nějakým čitelným způsobem. Používám tedy nahrazení

```
if(addWord.contains("_mezera_")){addWord = addWord.replace("_mezera_"," ");}
```

Druhá podmínka řeší správnou velikost prvních písmen při doplnění rozepsaného textu.

```
if (Character.isUpperCase>LastWord.charAt(0)) {
addWord = addWord.substring(0,1).toUpperCase().concat(addWord.substring(1));
}
```

### 4.3.3 Predikce následujícího písmene

Jak už bylo zmíněno jedná se o barevné zvýraznění až tří písmen s největší pravděpodobností výskytu na konci právě rozepsaného textu. Toto zvýraznění je obstarávané pomocí `setKeyPrediction()` změnou barvy pozadí tlačítek. Za touto metodou však stojí logika výběru tlačítek navržených k přebarvení. O tento výběr se stará `getFrequentChars()`, a to tak, že nalezne příslušný uzel trie, jehož řetězec odpovídá konkrétnímu stavu rozepsání posledního slova a ze všech jeho potomků pak hledá tři s největší hodnotou vrácenou metodou `getWordsInSubTree()`. Abych předešel nutnosti vždy pro každý uzel počítat tuto hodnotu a procházet větve stromu, mám v každém uzlu trie uchovaný aktuální počet slov v jeho podstromu.

#### 4.3.4 Přidání slov do slovníku

Nejprve bych se chtěl krátce věnovat tomu, jak poznáme, že uživatel dopsal slovo a můžeme ho vložit do slovníku. Děje se tak pomocí oddělovacích znaků jako jsou mezera, čárka, tečka, otazník nebo vykřičník. Tyto znaky budou zpravidla ohraničovat slovo z obou stran. `saveWordToDict(String word)` obstarává všechny záležitosti ohledně zjištění slova a kontroly jeho výskytu ve slovníku. Jak je implementováno samotné získávání posledního slova z textu už jsem se zmiňoval.

Pokus o uložení slova tedy nastává po každém použití oddělovacího znaku a pokud není řetězec prázdný dojde ke kontrole předchozího vložení. Není-li toto slovo slovníku známé, je vloženo do trie struktury metodou `add()` a zároveň i na konec souboru `general_cze.dat` obsahujícího slovník.

#### 4.3.5 Rozložení kláves

Při tvorbě layoutu jsem vycházel z těchto základních požadavků.

- Klávesy musí být velké
- Rozložení bude obsahovat všechny české znaky
- Rozmístění znaků bude založeno na jejich výskytu v jazyce

Rozhodl jsem se nejprve vytvořit layout daného počtu tlačítek v mřížce bez toho, abych určil nějakému znaku pozici. Základní mřížka byla 8x6 polí pro písmena. Vzhledem k počtu znaků a jistotě, že nejčastěji používaným znakem bude mezera, jsem její tlačítko posadil doprostřed a zvětšil jeho šířku na dvojnásobek. Jedno pole na okraji zůstalo prázdné. Na levé straně jsem umístil nápovědu slov ze slovníku, na spodní hranu prostor pro zobrazení napsaného textu a na pravou stranu ještě jeden sloupec tlačítek funkcí jako je mazání, enter, shift a NumPad obsahující další čísla a znaky.

Podle Fitts' Law [18] víme, že čas potřebný k přesunutí kurzoru a stisku tlačítka vypočítáme jako

$$MT = a + b \log_2 \left( \frac{D}{W} + 1 \right)$$

Čas nutný pro tuto akci je tedy možné ovlivnit změnou velikosti rozestupů středů tlačítek  $D$  a jejich šířkou  $W$ . Ostatní proměnné ve vzorci jsou empiricky určené. Z tohoto důvodu, jsem se rozhodl, že vzhledem k požadavku na velká tlačítka bych mezi nimi měl alespoň ponechávat minimální mezery.

Při rozmístění písmen jsem postupoval následovně. Do středu ke tlačítku mezery jsem přidal dva nejčastější znaky, kterými jsou E a O. Vznikla mi tak čtveřice usazených kláves. Nyní jsem měl 8 ke středu nejbližších prázdných polí. Pythagorovou větou lze jednoduše ověřit, že ostatní pole včetně těch diagonálně přisedlých na čtverec mají své středy ve vzdálenosti ještě větší. Do těchto polí jsem musel naplnit dalších osm nejfrekventovanějších písmen češtiny. Usazoval jsem tedy písmena A, N, T, S, I, V, L a R. Nyní už jsem však musel brát zřetel i na další faktor a tím je výskyt bigramů, tedy dvojic znaků. Vypsal jsem si ty nejčastější bigramy již umístěného O a E s právě usazovanými osmi písmeny.

Tabulka 4.1: Frekvence bigramů tvořených z O nebo E a písmen A,N,T,S,I,V,L,R.

Písmeno	Bigramy podle frekvence
O	ov, or, ot, on
E	en, et, el, er

Podle této tabulky jsem se snažil pokládat písmena obsažená v často se vyskytujících bigramech na co nejbližší se nacházející tlačítka. Proto jsou například O a V, písmena jednoho z nejčastějších bigramů hned vedle sebe. Samozřejmě, že vzhledem k velkému počtu bigramů a omezenému počtu tlačítek v okolí nejsou některé vzájemně na pozicích přímo sousedících. Tento proces ukládání písmen do nejbližšího okolí již usazených tlačítek jsem opakoval, dokud nebyla celá klávesnice zaplněna.

q	f	ě	í	á	é	g	d'
w	c	m	n	t	d	b	,
x	j	l	e	o	v	u	.
ň	ř	i	Mazec	s	h	!	
t'	ž	p	a	r	k	y	?
ó	ú	z	č	š	ý	ů	

Obrázek 4.3: Ukázka rozmístění znaků podle jejich pravděpodobnosti.

Výše uvedený obrázek se věnuje právě zmiňovanému rozmístění znaků na klávesnici. Platí pro něj, že čím výraznější je červené zbarvení tlačítka písmene, tím větší je jeho frekvence použití v českém jazyce. Jak je vidět, nejvíce používané znaky mají své místo blízko středu celé klávesnice.



## Kapitola 5

# Testování

Nedílnou součástí celého procesu tvorby softwaru musí být i jeho testování. Testy budeme provádět z důvodu ověření funkčnosti, zjištění případných nedostatků nebo nedokonalých řešení. Cílem je získat jiný pohled na námi řešený problém a využití poznatků k dalšímu zlepšení. Zároveň je také potřeba sesbírat data o kvalitě programu a zhodnotit co je přínosem a co nedostalo našemu očekávání. Získané informace nám také umožní porovnat naši aplikaci s již existujícími produkty. Tyto testy také pomohou posoudit výsledek práce jako celku. Nejprve bych se rád zmínil o testech, které jsem prováděl sám a následně rozeberu testování s uživateli.

### Funkční testy

Tyto testy se týkají hlavně prediktivní klávesnice. Bylo u ní potřeba zjistit, zda všechny důležité funkce poskytují správný výstup. Některé testy jsem vytvořil již při psaní kódu trie struktury, abych měl jistotu, že budou slova vkládána a vybírána správně. Pomocí sad testovacích slov jsem plnil vstup a v debug modu sledoval práci algoritmu. Pomocí stejné metody jsem testoval i funkčnost zobrazení všech vhodných slov ze slovníku a správné vyhodnocení nápovědy dalších nejpravděpodobnějších znaků. Mé pozornosti neuniklo ani chování event listenerů v nestandardních situacích jako zvolení prázdného pole nápovědy nebo použití dvojkliků při jejím výběru. Dále jsem při samotném psaní zadával rozličné kombinace znaků, mezer, nových řádků a zkoumal, jak si s nimi poradí algoritmy nápověd a mazání slova.

### Test přesnosti a rychlosti

Jedním z dalších testů, který mi pomohl při finalizaci softwaru byl test přesnosti a rychlosti pohybu kurzoru. Vždy jsem zvolil dostatečně malý objekt, například tlačítko minimalizace okna. Vyzkoušel jsem rychlost kurzoru rychlým přejetím přes celou obrazovku a pak se snažil zastavit kurzor na středu tohoto tlačítka. Tento test byl postupně zkoušen s různým nastavením. Účelem bylo optimalizovat rychlost pohybu kurzoru a upravit velikost klidové zóny tak, aby byla zároveň zaručena i přesnost.

## 5.1 Testování s uživateli

Když jsem nabil dojmu, že blok funkčních testů byl důkladně proveden, začal jsem připravovat potřebné materiály k testu s uživateli, shánět potencionální participanty a řešit místo, kde se bude testovat. Během této doby jsem také provedl pilotní test, který do značné míry ovlivnil vzhled úkolů. Z časových důvodů jsem se rozhodl koncipovat testy spíše jako kvalitativní a vybrat pět participantů. I při tomto menším počtu účastníků by však měl test odhalit téměř 90% problémů použitelnosti.

Od počátku bylo rozhodnuto, že vzhledem k náročnosti shánění uživatelů s postižením proběhne test s lidmi, u kterých budeme toto postižení horních končetin simulovat. Během celého testu tedy měli participanti své ruce volně v klíně a jejich používání jim bylo striktně zakázáno. Participanty jsem vybíral tak, abych měl zástupce obou pohlaví. U žen jsem vybral několik studentek ergoterapie a to hlavně z důvodu znalosti problematiky a jejich dřívějších zkušeností s podobnými systémy pro postižené. Předpokládal jsem, že jejich znalosti mi pomohou lépe zhodnotit přínos tohoto programu.

### 5.1.1 Výběr úkolů

Při výběru úkolu jsem se zaměřil na každodenní činnosti vykonávané na počítači. Vybrány byly ty, se kterými se dle mého názoru budou uživatelé s handicapem s největší pravděpodobností potýkat. Jednou z priorit při výběru úkolů bylo použití zásadních součástí tohoto projektu. Již předem jsem se tedy rozhodl, že kromě psaní budou úkoly vyžadovat i práci s oběma tlačítky myši a drag and drop.

Kdybych měl obecně říci, které činnosti jsou v testech vyzkoušeny

- Nastavení polohy hlavy a pohyb kurzorem
- Prohlížení webové stránky
- Psaní e-mailu
- Psaní textu na dvou rozdílných klávesnicích
- Práce se soubory a složkami

Nyní se pojdme podívat na jednotlivé úkoly, rozebrat na co jsou zaměřené, co během nich chceme sledovat a jaké výsledky očekáváme. Úkolů bylo vytvořeno šest a jejich délka a složitost byla upravena tak, aby se celkový čas testů pro participanty pohyboval okolo jedné hodiny. To se zdá být ideální délkou co se týče udržení pozornosti, zájmu a koncentrace na řešení problému, sdělování pocitů a hledání chyb.



### Úkol č.1 - Prvotní nastavení polohy hlavy a pohyb kurzorem

*Pomocí tónu zkalibrujte kurzor myši do středu obrazovky. Pohybem hlavy dostaňte kurzor nejdříve na levý a následně na pravý okraj obrazovky. Poté najedte i na horní a dolní.*

První úkol je ve své podstatě velmi jednoduchý a slouží spíše k seznámení participanta s pohybem kurzoru pomocí své hlavy. Zároveň se v něm uživatel setká s prvním zvukovým gestem a tím je kalibrace pozice hlavy. Zde budu pozorovat, zda si před samotnou kalibrací uživatel posune obličej do blízkosti středu záběru.

### Úkol č.2 - Prohlížení webových stránek

*Otevřete webový prohlížeč Opera (Nachází se na ploše). Pomocí klávesnice napište adresu www.idnes.cz a zkopírujte ji. Do prohlížeče ji následně vložte pravým tlačítkem myši -> vložit a přejít na. Přejděte do rubriky kultura a otevřete její první článek. Sjedte úplně dolů na konec článku.*

Tento úkol je už o něco složitější. Snažím se s ním ověřit použitelnost programu k prohlížení webových stránek. Opět v něm bude otestován pohyb kurzoru, ale tentokrát i přesnost, s kterou se lze zaměřit na odkazy nebo scrollovací lištu. V úkolu je také vyžadován dvojklik, díky kterému ověřím, zda-li systém rozpoznání gest nemá problém s rychle za sebou následujícími gesty. Dále je potřeba používat i gesto stisknutí pravého tlačítka. Budu tedy sledovat, jak jednoduché je pro uživatele střídavě vydávat vysoké a nízké tóny.

### Úkol č.3 - E-mail

*V Opeře otevřete www.seznam.cz. Na stránce se přihlaste na účet a klikněte na nová zpráva. (Přihlašovací údaje i e-mail adresáta máte na zvláštním papíru) Odešlete e-mail s předmětem "Test" a obsahem "Ahoj, ve středu v 11:25 ti pomohu s testováním."*

Tento úkol je zaměřen na ověření použitelnosti při online komunikaci. Do jisté míry je podobný úkolu předcházejícímu. Využívají se při něm podobné úkony, ale je zaměřen spíše na potřebu střídavého psaní textu na klávesnici a práce s kurzorem ve webovém prohlížeči. Zároveň se jedná o první úkol, kde se již více využije klávesnice a to včetně psaní čísel.

### Úkoly č.4 a č.5 - Psaní textu

*Pozorně si přečtěte tyto věty a přepište je do poznámkového bloku. "Můj bankovní účet je přečerpán. Sloni se bojí myši. Uveďte prosím své datum narození." Stejný text, který byl v minulém cvičení přepište do poznámkového bloku za použití součásti Windows - klávesnice na obrazovce.*

Tyto dva úkoly byly vytvořeny z důvodu otestování psaní na vytvořené klávesnici. Jejich úkolem je sledovat rychlost psaní a frekvenci překlepů a případně odhalit jiné potencionální problémy. Dohromady nám oba úkoly poskytnou porovnání výkonnosti a použitelnosti vytvořené klávesnice vůči té od Microsoftu. Budeme při nich sledovat, jak uživateli pomáhají nápovědy dalších znaků, slov ze slovníku a zda si uživatel zvyká na jiný layout.

## Úkol č.6 - Práce se soubory a složkami

*Vytvořte adresář a pojmenujte ho Zvířátka. Uvnitř této složky přes pravé tlačítko -> nový -> textový dokument vytvořte soubor slon.txt. Následně soubor přetáhněte kurzorem na plochu.*

Tento úkol testuje další každodenní situaci, ve které se může uživatel nalézt. Opět sledujeme přesnost práce s kurzorem, využití zvukových gest tentokrát včetně držení levého tlačítka. Budeme se soustředit na to, jak si uživatel poradí s jeho uvolněním. A samozřejmě jako vždy při psaní sledujeme, jak uživatel pracuje s klávesnicí.

### 5.1.2 Testovací podmínky

Všechny testy byly provedeny na notebooku Lenovo Z580A se čtyřjádrovým procesorem Intel i7 2.1GHz, grafickou kartou NVIDIA GeForce GT 640M (2GB) a 8GB RAM DDR3. Program byl pouštěn pod operačním systémem Windows 7 a k ovládání byla využita integrovaná webová kamera Lenovo EasyCamera. Uživatelům byla poskytnuta sluchátka s mikrofonom, aby byl eliminován případný hluk z okolí. Dalšími faktory, které testy ovlivňovaly a nebyly pro všechny testy stejné, jsou denní doba a s tím spojené osvětlení záběru. Podobně také barevnost pozadí za uživatelem, barva jeho oblečení nebo barvy vyzářené monitorem při velkém jasu v temnějším prostředí.

### 5.1.3 Průběh testování

Všichni účastníci byli před testem krátce seznámeni se vzhledem a s jednotlivými okny aplikace. Každému z nich byla nakalibrována hodnota hranice mezi vysokým a nízkým tónem. Každému z účastníků byl předán k vyplnění jeden dotazník před testem, seznam úkolů a jeden post testový dotazník.

#### Participant 1

Prvním testujícím byla mladá studentka ergoterapie. Podle dotazníku se o osoby s postižením zajímá a setkala se, jak s lidmi s postižením horních končetin, tak s lidmi s psychickým postižením, které může výrazně znesnadnit ovládání počítačových periferií. Již dříve pracovala, jak s virtuální klávesnicí od Microsoftu, tak s alternativními metodami ovládání počítače. A to konkrétně se sledováním lidského oka a ovládáním pomocí reflexního bodu na čele. Jako nevýhody těchto metod uvedla samovolný pohyb brýlí na obličeji a slepování tečky z čela. Od testovaného programu očekávala, že bude fungovat lépe než drahé komerční metody a usnadní postiženým práci s počítačem. Doufala v odhalení nedostatků produktu, nabití zkušeností a praxí do budoucna. Nyní se dostáváme k průběhu jednotlivých testů.

**Úkol 1** byl splněn v časovém úseku 1:15 bez sebemenších problémů. Uživatel si při kalibraci správně nastavil obličej do středu záběru.

**Úkol 2** byl splněn v celkovém čase 5:47. Z toho 2:34 trvalo uživateli napsat své první slovo respektive celou webovou adresu. Během psaní byl nalezen jen jeden problém a to když v 2:18 po začátku testu participant hledal na layoutu písmeno Z. Zbylé úkony zabraly tyto časy: 1:31 kopírováním textu z klávesnice a jeho vložením do prohlížeče a 1:42 navigací na stránce, při níž se uživateli v jednu chvíli nedařilo nalézt kurzor schovaný u okraje obrazovky. Žádný

problém s rozpoznáním dvojkliku ani s přesností při použití scrollbaru nenastal. Uživatelé nečinilo problém ani ovládání nízkými a vysokými tóny.

**Úkol 3** byl ukončen úspěšným odesláním e-mailu po 24 minutách. Uživatel se během psaní WWW adresy překlikl a doplnil ze slovníku omylem špatné slovo. I přes nutnost opravy mu tento úkon zabral 1:33. V 5 minutě 35 vteřině při vkládání uživatelského jména omylem vyvolal zvukem stisk tlačítka. Dále pak při přihlašování uživatel přešel tlačítko přihlásit a musel se navracet zpět. S následnou minimalizací okna už ovšem problém neměl. Přihlášení do e-mailu tedy proběhlo za 9:10. Při psaní adresy příjemce došlo k jednomu překlepu a napsání jednoho znaku navíc z důvodu kýchnutí. Zavináč byl nalezen bez problému v NumPadu. Další průběh úkolu byl bez komplikací. Tělo zprávy bylo napsáno za 4:56.

**Úkol 4**, tedy psaní na vytvořené klávesnici trval 12:15. Na začátku psaní první věty došlo k jednomu překlepu. Uživatel jedenkrát využil nápovědu slov a za 4:37 tedy napsal první větu. Ze sledování uživatele bylo znát, že jeho orientace v layoutu se po každé větě zlepšuje. Druhá věta byla dopsána v čase 7:25 a třetí a poslední věta byla dopsána 11:09 a to i přes jeden překlep a napsání jedné mezery navíc. Zbýlý čas zabralo vložení textu do poznámkového bloku.

**Úkol 5**, tedy napsání stejného textu na klávesnici Windows trval 15:06. První věta byla napsána za 6:17. Hned první stisk na klávesnici skončil překlepem. Celkově jich bylo v první větě šest a u některých znaků na ně uživatel najel například až napodruhé. Druhá věta byla dopsána v čase 9:39 pouze s jedním překlepem. Ve třetí větě se vyskytly celkem tři překlepy. Participant poukázal na obtížnost při rychlém pohybu kurzorem přesně stisknout malé tlačítko.

**Úkol 6** byl participant schopen vyřešit za 6:32. Jednou v počáteční fázi úkolu byl nucen rekalibrovat rozpoznávání obličejů, aby mohl snáze jet kurzorem nahoru. Jiný problém při řešení nenastal a uživatel i bez jakéhokoli váhání zvládl uvolnit levé tlačítko z módu jeho držení.

Participant ohodnotil přesnost pohybu kurzoru slovem výborná. Jeho rychlost pak slovem dobrá s poznámkou, že šlo o to se naučit ji kontrolovat. Při testech uživatel neměl problém s rozpoznáním houkání a byl velmi spokojen s nápovědou znaků a slov. Největší přínos prvků u klávesnice viděl v tomto pořadí: layout znaků, nápověda dalšího znaku, nápověda slov a na posledním místě velikost znaků. Při dotazu na porovnání obou použitých klávesnic sdělil, že má každá své výhody i nevýhody, ale že jsou obě dobře použitelné. Zaujalo ho, že lze tímto způsobem nahradit drahé systémy pro postižené za desítky tisíc.

## Participant 2

Dalším participantem byl mladý absolvent střední průmyslové školy. Jediné jeho předchozí setkání se systémem použitelným pro postižené je psaní na virtuální klávesnici ve Windows. Od testovaného programu očekával, že s jeho pomocí budou lidé s postižením více zapojeni do světa techniky. U uživatele byl zpočátku problém s nastavením hraniční hodnoty tónů. Rozdíl mezi zvuky, které vydával jako vysoké a nízké tóny, byl totiž velmi malý.

**Úkol 1** byl splněn v čase 0:58 bez nejmenších problémů. Uživatel si při kalibraci ponechal obličej blíže k hornímu okraji obrazovky.

**Úkol 2** byl splněn v celkovém čase 5:15. Při psaní webové adresy došlo k jednomu překlepu, i přesto byl čas psaní 2:28. S přesností neměl uživatel žádný problém, scrollování

prováděl klikáním na lištu místo tažení jezdce. Uživatel měl několikrát během testu problém vydat vysoký tón.

**Úkol 3** byl ukončen úspěšným odesláním e-mailu po 19 minutách 59 vteřinách. Uživatel hned od počátku dokázal využívat nápovědu slov. Adresu stránky tím pádem napsal za 1:23. V 2 minutě 20 vteřině chtěl docílit větší rychlosti pohybu kurzoru, ovšem vyjel obličejem nad horní okraj záběru kamery. Dále pak při psaní přihlašovacích údajů musel jedenkrát recalibrovat rozpoznání obličeje, aby mohl pohodlně vyjet k hornímu okraji. Po 10 minutách úkolu se uživatel přihlásil na e-mail a začal vytvářet zprávu. Zavináč potřebný k napsání adresy byl nalezen bez problému. Při psaní obsahu zprávy byl třikrát použit slovník, nedošlo k jedinému překlepu a celkový čas psaní byl pouze 3:14.

**Úkol 4**, tedy psaní na vytvořené klávesnici, trval 10:11. V první větě se uživatel překlepl při výběru ze slovníku a doplnil slovo "bankovních" místo "bankovní". Chybu pohotově opravil smazáním posledních dvou písmen. Následně nastal ještě jeden překlep a to výběr písmena c místo j, větu dopsal za 3:11. Druhá věta byla dopsána v čase 5:59 opět s použitím slovníku a v té poslední došlo k jednomu překlepu a hojněmu využití slovníku. Dopsáno bylo v čase 9:25 text byl vložen do poznámkového bloku 10:11.

**Úkol 5**, tedy napsání stejného textu na klávesnici Windows, trval 16:18. První věta byla napsána za 5:26. Během psaní této věty došlo k sedmi překlepům. Zároveň ke konci psaní byla potřeba restartovat aplikaci rozpoznání zvuku. Po testu bylo zjištěno, že při zaslání znaku z Windows klávesnice do okna rozpoznání zvuku občas dochází k jeho zamrznutí. Čas restartu byl od výsledného času úkolu odečten. Druhá věta byla zvládnuta v čase 9:35 a při jejím psaní se vyskytly 3 chyby. Uživatel si stěžoval na malou velikost písmenek. Během poslední věty napsal čtyři překlepy a jednou byl nucen kalibrovat ovládání kurzoru. Prakticky vzato tedy udělal participant v každém slově jednu chybu.

**Úkol 6** byl participant schopen vyřešit za 5:29. Při pohybu k hornímu okraji byl rozpoznán krk místo obličeje a kurzor se pohyboval nahoru pomalu. Po recalibraci už problém nenastal. Dvakrát se uživateli stalo, že přešel malou nabídku. Nejprve možnost vložit při přejmenování a pak samotný txt soubor ve složce se zobrazením typu seznam. S uvolněním levého tlačítka z módu držení problém nebyl.

Participant ohodnotil přesnost a rychlost pohybu kurzoru slovem dobrá. Nápověda znaků a slov byla podle jeho slov výborná a hojně ji také využíval. Pohodlněji se mu psalo na nově vytvořené klávesnici a to hlavně z důvodu větších tlačítek. Rozložení kláves mu přišlo dobré, jen poukázal na dlouhodobý návyk na QWERTZ layout. Nejvíce přínosné mu přišly prvky v pořadí nápověda slov, nápověda znaku, rozložení kláves a velikost písmen. Na programu ho zaujal nápad a inovativní rozložení klávesnice. Díky jeho podnětu bylo rozhodnuto o zobrazení hraničních os pro pohyb kurzorem v okně zobrazujícím rozpoznání obličeje.

### Participant 3

Testů se zúčastnila také druhá studentka oboru ergoterapie na 1.lékařské fakultě. Její znalosti ohledně problematiky postižených i využití alternativních systémů ovládání byly také velkým přínosem. Již dříve měla možnost vyzkoušet ovládání pomocí pohybů oka, se kterým měla podle svých slov problémy kvůli oční řasence. Od produktu očekává, že pomůže lidem s omezenou hybností horních končetin ovládat efektivně počítač. Od testu očekává, že pomůže

odhalit případné skryté nedostatky. Uživateli musela být přenastavena hraniční hodnota tónů, protože nebyl schopen vydat tak hluboký tón.

**Úkol 1** byl splněn v čase 2:07. Žádné komplikace se neobjevily. Uživatel kalibroval s obličejem správně ve středu záběru. Zdálo se, že uživatel s kurzorem jezdí poněkud pomaleji.

**Úkol 2** byl splněn v celkovém čase 6:03. Při psaní webové adresy byl uživatel obezřetný, pohyboval kurzorem pomaleji a adresu dopsal 2:51. S přesností při zmiňované menší rychlosti problém nebyl. Scrollování provedl tažením jezdce. Uživatel musel během úkolu jedenkrát kalibrovat facetracker.

**Úkol 3** byl u třetího participanta splněn za 30 minut 57 vteřin. Uživatel měl pomalejší pohyb kurzorem, za to prakticky bez chyb a překlepů. Adresu stránky napsal za 2:44. Při psaní přihlašovacího jména jednou využil nápovědy slov. Vložení jména do příslušné kolonky na webu se mu podařilo až napodruhé a tak proběhlo přihlášení v čase 11:25. Ještě jednou se uživateli stalo, že přešel přes tlačítko, které chtěl stlačit, a to přes "Nová zpráva". Obsah zprávy byl napsán za 8:58. Uživatel se za celou dobu testu neodvážil pohnout kurzorem vyšší rychlostí a zároveň prakticky vůbec nevyužil možnosti nápovědy. Proto byl jeho čas znatelně pomalejší.

**Úkol 4** s vytvořenou prediktivní klávesnicí trval participantovi 17:11. V první větě došlo k stisknutí přebytečného znaku nechtěným vydáním zvuku. Bylo vidět, že již uživatel některé znaky nemusí hledat. Věta byla dopsána za 6:58. U dalších vět již uživatel používal nápovědu slov. Druhá a třetí věta pak byla napsána v časech 11:29 a 15:55. Uživatel se naučil uvést kurzor do vyšší rychlosti.

**Úkol 5**, tedy napsání stejného textu na klávesnici Windows, trval 20:29. První věta byla napsána za 9:17 a bylo při ní uděláno 7 překlepů. Věta druhá byla napsána v čase 13:16 s dvěma chybami a poslední část textu v celkovém čase úkolu, tedy 20:29. V poslední větě musely být postupně opraveny 4 překlepy. Uživatel si po celý čas stěžoval na malou velikost kláves klávesnice Windows a zmínil také nevýhodu, že neobsahuje slovníku.

**Úkol 6** byl participant schopen vyřešit za 7:35. Uživatel měl během úkolu problém pouze s přejetím z nabídky "Nový" na volbu "Složka" nebo "textový dokument". S uvolněním držení levého tlačítka problém nebyl. Psaní na klávesnici probíhalo bez komplikací.

Po testu se participant rozpovídal. Přesnost a rychlost pohybu kurzoru hodnotil jako výbornou a i s nápovědami slov a znaků byl velice spokojen. Celkově klávesnici hodnotil velmi pozitivně, protože psaní na ní bylo podle jeho slov jednodušší, rychlejší a s menším počtem chyb oproti srovnávané klávesnici z Windows. Jiné rozložení znaků mu nevadilo, prý jde jen o zvyk. Pozitivní přínos součástí viděl v pořadí velikost písmen, nápověda slov, nápověda dalšího znaku a jiné rozložení znaků. Při poslední otázce co ho nejvíce potěšilo, zklamalo a zaujalo zmínil, že program výborně reagoval na hlasové povely.

#### Participant 4

Další účastníci testů byla opět žena. Osobně zná několik postižených a o jejich problematiku se zajímá, ale nemá předchozí znalosti s žádným programem ovládajícím počítač tohoto typu, ani s žádnou virtuální klávesnicí. Od programu očekává, že pomůže postiženým ovládat počítač a umožní jim tak s jeho pomocí dělat spoustu věcí. Těší se, že při testu zjistí jak tyto systémy pro handicapované fungují.

**Úkol 1** byl splněn v čase 1:25. Uživatel měl obličej správně ve středu obrazovky a neměl s kalibrací ani s následným pohybem kurzoru problémy.

**Úkol 2** byl dokončen ve výborném čase 4:02. Při psaní webové adresy se uživatel dopustil jedné chyby, kterou však okamžitě opravil. Adresu měl dopsanou 2:19. S přesností kurzoru nebyl problém, byl jen menší problém při vkládání adresy do webového prohlížeče. Došlo k přejetí prostoru pro vložení URL. Scrollování bylo provedeno klikáním na spodní okraj scrollovací lišty.

**Úkol 3**, zabývající se elektronickou komunikací, byl u tohoto participanta splněn za 21 minut a 3 vteřiny. Adresa stránky byla napsána za 2:45. S pohybem kurzoru ve směru dolu měl uživatel chvíli problém. Po pár vteřinách pohybování hlavou se poupravila pozice rozpoznávaného obličeje a problém sám odezněl. Při psaní přihlašovacího jména a hesla byla jednou využita nápověda slova "test". Přihlášení pak proběhlo v čase 7:52. S psaním předmětu uživatel neměl potíže, za to při jeho vkládání dvakrát přešel volbu vkládající obsah schránky na danou pozici. Ještě jednou se participant setkal s problémem pohybu kurzoru. Jeho krk byl rozpoznán namísto obličeje, ale jeho zakrytím se vše vrátilo do funkčního stavu. Obsah zprávy byl napsán za 5:47. Uživatel psal bez chyb, ale pohyb kurzoru byl pomalejší.

**Úkol 4** s klávesnicí vytvořenou pro tuto práci trval participantovi 13:33. Během plnění tohoto úkolu nedošlo k jedinému překlepu uživatele. První ze tří vět byla dopsána za 5:47. Druhá a třetí věta pak byla napsána v časech 8:28 a 12:51. Uživatel použil celkem čtyřikrát nápovědu slova ze slovníku a jednou musel kalibrovat pozici své hlavy. K žádným závažnějším problémům však nedošlo.

**Úkol 5**, tedy přepisování textu ze čtvrtého úkolu na klávesnici Windows, trval 18:31. U menších tlačítek docházelo častěji a mnohem výrazněji k přejetí požadované pozice. První věta byla napsána za 7:47 a byl při ní udělán 1 překlep. Věta druhá byla napsána v čase 12:19 s jednou chybou a v poslední větě nedošlo k chybě. Uživatel neudělal chyb mnoho, za to ale strávil hodně času zastavováním kurzoru na malých tlačítkách.

**Úkol 6** orientovaný na soubory a složky byl participant schopen vyřešit za 6:42. Uživatel měl jen menší problémy s přejetím malých nabídek a s uchopením souboru při snaze ho přemístit. Špatným počátečním umístěním kurzoru vyvolával místo uchopení akci označení více souborů. S uvolněním držení levého tlačítka problém neměl a psaní na klávesnici probíhalo obdobně bez komplikací.

Po testu participant sdělil, že přesnost a rychlost pohybu kurzoru je dobrá, jen je potřeba si zvyknout na setrvačnost jeho pohybu. Zároveň prozradil, že měl pocit, že při pohybu hlavou do stran hýbe současně tělem i hlavou dopředu/dozadu. S nápovědami slov a znaků klávesnice byl velice spokojen a jiné rozložení znaků mu nepůsobilo problémy. Pozitivní přínos součástí viděl v pořadí: nápověda slov, nápověda dalšího znaku, rozložení znaků a velikost tlačítek. Uživatele při používání velmi potěšila nápověda dalšího znaku, protože mu mnohokrát usnadnila hledání písmen na neznámém layoutu.

## Participant 5

Posledním účastníkem testu byl inženýr pracující v oblasti vývoje softwaru. Jeho předchozí znalosti zahrnují používání virtuální klávesnice ve Windows. Od programu očekává, že bude

handicapovaným lidem schopem kvalitně nahradit hardwarové periferie a že jim umožní samostatně pracovat s počítačem. Jeho snahou bylo pomoci odhalit co nejvíce chyb a nedokonalostí použitých řešení.

**Úkol 1** byl splněn v čase 1:05 na druhý pokus. Při prvním spuštění a prudkém pohybu hlavy na okraj záběru byla knihovnou OpenCV zajišťující facetracking vyhozena výjimka. S kalibrací ani s následným pohybem problém nebyl.

**Úkol 2** testující navigaci na webových stránkách byl splněn v čase 4:59. Při psaní webové adresy uživatel udělal jednu chybu. Psaní této adresy zabralo 2:02. Menší problém se objevil až při vkládání adresy do webového prohlížeče, došlo k přejetí možnosti vložit. Scrollování bylo provedeno klikáním na spodní okraj scrollovací lišty. V jednu chvíli došlo k chybnému rozpoznání oranžově zabarveného rámu okna namísto obličeje. Opakovaným pohybem hlavy se elipsa rozpoznání vrátila na uživatelův obličej.

**Úkol 3**, tedy psaní e-mailu, byl u tohoto participanta splněn za 20 minut a 28 vteřin. Adresa stránky byla napsána za 2:16. Při psaní přihlašovací jména došlo k jednomu překlepu. Přihlášení pak úspěšně proběhlo v čase 7:50. S psaním předmětu uživatel neměl potíže, ale při snaze označit pro něj určené pole a text do něj vložit byl vydaný hlasový povel příliš tichý a nedošlo tak k jeho rozpoznání. Obsah zprávy byl napsán za 5:23. Uživatel psal bez chyb, jednou využil slovník a zajímavostí bylo, že kurzorem pohyboval v obloukových křivkách místo nejkratší přímé trajektorie.

**Úkol 4**, s nově vytvořenou klávesnicí, trval participantovi 10:18. Během plnění tohoto úkolu došlo k dvěma překlepům, kdy se uživatel snažil trefit okraj tlačítka a pokračovat bez zastavení pohybu kurzoru. První ze tří vět byla dopsána za 4:05. Druhá věta byla pak v čase 6:17 a ta poslední 9:50. Nápověda slova ze slovníku byla využita třikrát.

**Úkol 5**, tedy přepisování stejných vět jako v minulém úkolu, ovšem tentokrát na klávesnici z Windows, trval 14 minut a jednu vteřinu. Uživatel výrazněji přejížděl tlačítko, na kterém chtěl změnit směr pohybu. První věta byla napsána za 5:28 a byly při ní udělány 3 překlepy. Věta druhá pak byla kompletní v čase 8:50 s jedinou chybou. V poslední větě k chybě nedošlo a úkol byl úspěšně dokončen.

**Úkol 6** se soubory a složkami byl participantem vyřešen za 6:24. Uživatel nejprve dal přejmenovat složku a pak až si uvědomil, že musí nejprve napsat text. Následně při psaní názvu složky jedenkrát chyboval. Při snaze vyvolat akci držení levého tlačítka omylem použil gesto pro kalibraci. S následným uvolněním drženého tlačítka problém nenastal.

Při závěrečném pohovoru participant řekl, že na přesnosti kurzoru mu vadí jeho přejíždění. S nápovědami slov klávesnice byl spokojen. Jiné rozložení znaků mu nevadilo, ale měl pocit, že se mu na něm píše pomaleji. Na klávesnici mu chyběla zvuková zpětná vazba. Nejvíce ocenil nápovědu dalšího znaku, která mu často vypomohla. Dále pak viděl největší přínos ve slovníku, velikosti písmen a na poslední místo dal odlišné rozložení znaků. Zajímavostí bylo, že tento participant jako jediný používal citoslovce "ha" namísto "hmm".

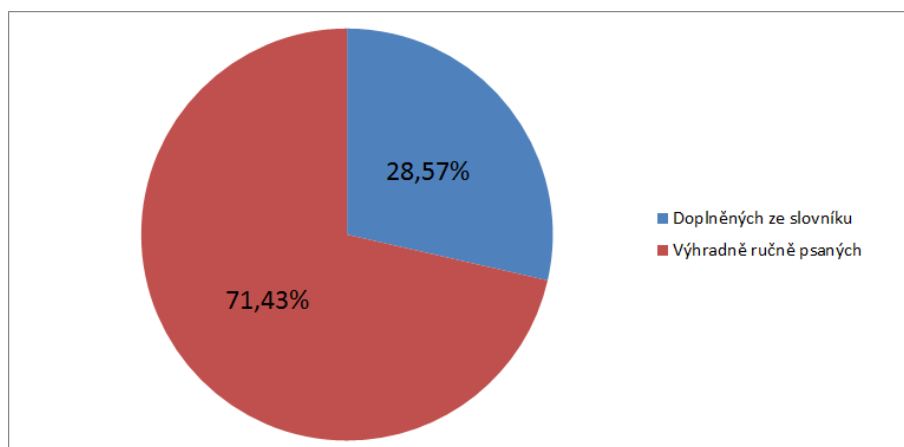
#### 5.1.4 Výsledky testování

Co tedy můžeme z testů usuzovat? I přes některé nedostatky výsledky naznačují, že by tento program mohli handicapovaní používat k běžné práci na počítači. Všichni účastníci byli v první řadě schopni kompletně dokončit jim zadané úkoly. I přes to bylo nalezeno několik nedostatků, které zde uvedu v pořadí podle jejich důležitosti:

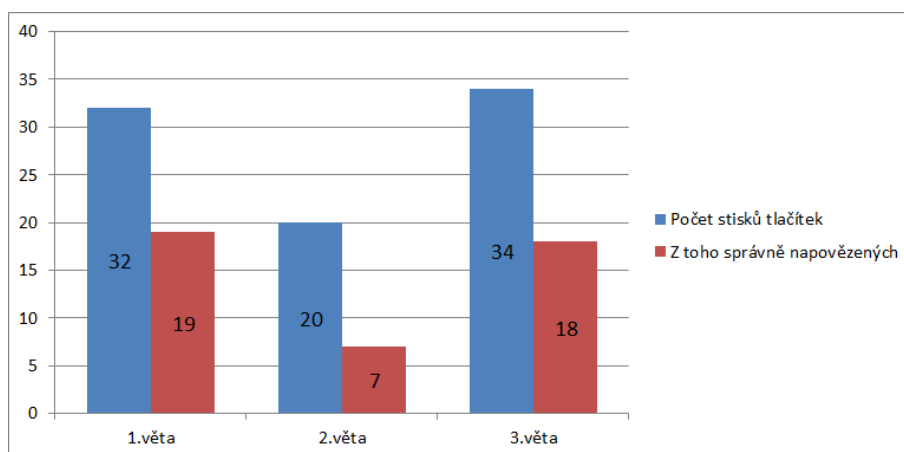
- Náchylnost rozpoznání obličeje na vnější vlivy. Obličej je v některých nevhodných světelných podmínkách rozeznán chybně nebo nepřesně. Faktory, které mohou tuto chybu zapříčinit jsou například: oblečení, pozadí za osobou nebo barevné ozáření obličeje jasným monitorem. Někdy též dochází k rozpoznání krku místo obličeje, což znepříjemňuje pohyb kurzoru v ose Y.
- Zamrznutí aplikace rozpoznání neverbálních gest. Tento problém nastává pouze při použití s klávesnicí Windows a je vyvolán stlačením znaku na klávesnici při aktivním okně rozpoznání zvuku. Použitím klávesnice z bakalářské práce se lze tomuto problému kompletně vyhnout.
- Nepřesnosti při manévrování kurzorem na velmi malých plochách. Myslím si, že jde z velké části o odhad a zvyk na ovládání pohybem hlavy. Jde o to si osvojit jemné a přesné pohyby hlavou. Nejvíce podobných problémů nastalo při psaní na klávesnici, která je součástí Windows. Při použití její alternativy z této práce už zbývá jen pár nepřesností při minimalizacích oken, vytváření souborů a podobně.
- Závislost na hlasových schopnostech jedince. Jeden z participantů měl menší problém s nastavením hranice pro rozpoznání vysokého od nízkého tónu. Lze si tedy představit, že se může vyskytnout osoba, pro kterou bude její hlas ještě větší překážkou v tomto ovládání. Většiny osob se však tento problém týkat nebude.
- Nejasnost polohy bodu, od kterého se změní směr pohybu kurzoru. Ke změně tohoto bodu dochází při každé kalibraci a jeho znalost nám umožňuje odložit kurzor do klidového bodu a lépe předvídat pohyb kurzoru. Bylo jednoduše vyřešeno zobrazením os pro určení směru pohybu.
- Nedostatečná zpětná vazba klávesnice při stlačení tlačítka kopírovat. Při kliknutí hlasovým gestem byla zpětná vazba vidět nepostřehnutelně krátký okamžik. Proto byla zpráva o úspěšném kopírování předělána a nyní se zobrazuje až do stažení kurzoru z povrchu tlačítka. Zároveň byla na základě podnětu od jednoho participanta dodělána i zvuková zpětná vazba tlačítek.
- U rozpoznání hlasových gest je menším nedostatkem i to, že uživatel nemůže "houkat" potichu a zároveň je omezen i jakýmkoli větším hlukem z okolí, který může do jeho ovládání zasáhnout.

Naopak asi největší úspěch jsem zaznamenal u mé klávesnice. Kladně byly hodnoceny hlavně její prediktivní funkce. Jako nejužitečnější byla označena nápověda slov. Jak často byla participanty využívána můžete vidět na obrázku 5.1. Dále bylo chváleno zvýraznění dalšího znaku, jiné rozložení kláves a velikost tlačítek. Žádný z uživatelů se nevyjádřil záporně k tomuto uspořádání. Zároveň u všech účastníků bylo možno sledovat, jak se časem zlepšují v orientaci mezi znaky.





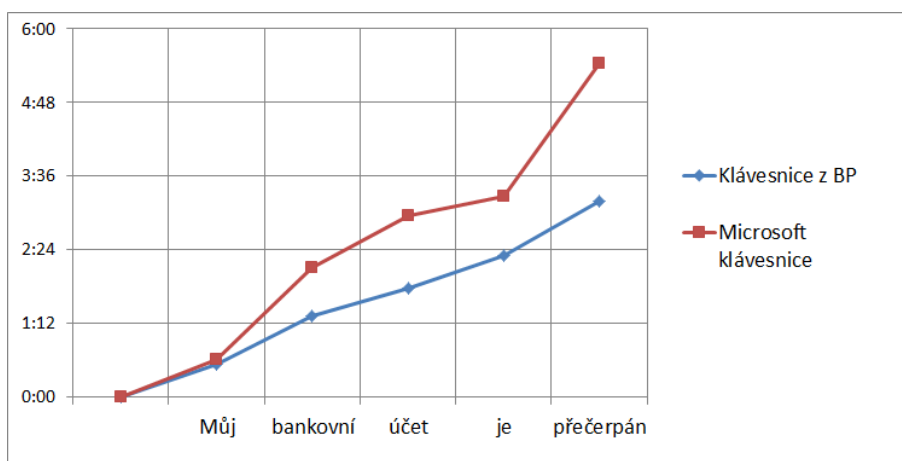
Obrázek 5.1: Průměrný podíl slov doplněných ze slovníku ve 4.úkolu .



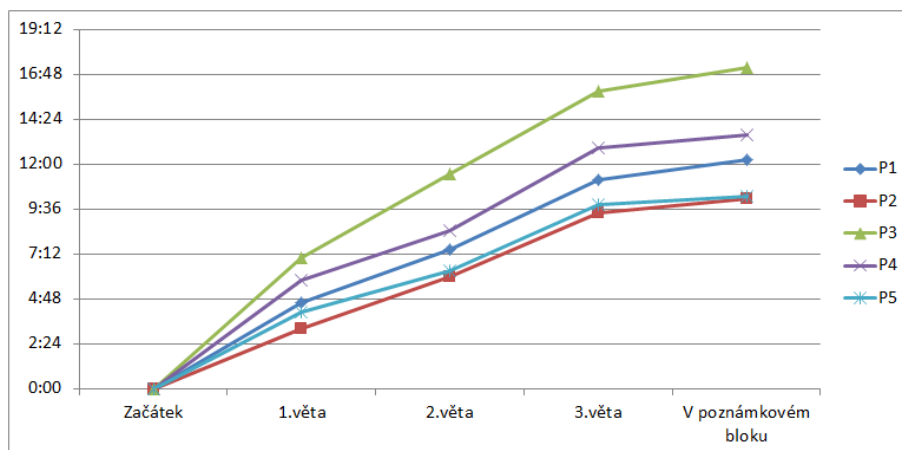
Obrázek 5.2: Ukázka efektivity predikce dalšího znaku v úkolu č.4

### Porovnání klávesnic

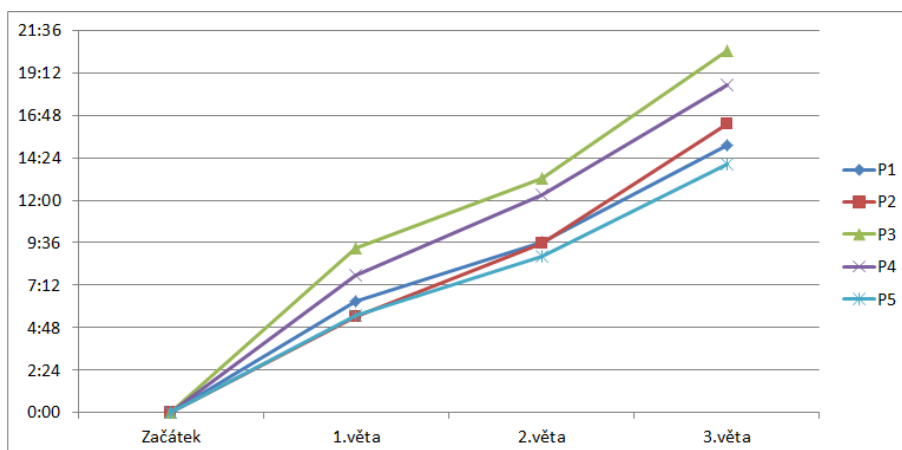
V komparativním testu dosáhla klávesnice z této práce v porovnání s virtuální klávesnicí z Windows výborných výsledků. Byl u ní naměřen menší počet chyb a kratší čas potřebný k psaní. Průměrný rozdíl času, naměřený při psaní tří vět, byl mezi klávesnicemi 24,8% ve prospěch té prediktivní. Velikost kláves z pohledu rychlosti naopak nahrává spíše klávesnici z Windows, ale na jejích menších tlačítkách si musel prakticky každý z účastníků vybrat mezi pomalým pohybem kurzoru nebo nutností opravovat své chyby vzniklé z překlepů při větší rychlosti. Čas ušetřený na nutnosti trefit malé tlačítko a opravovat časté chyby se nakonec ukázal jako rozhodující.



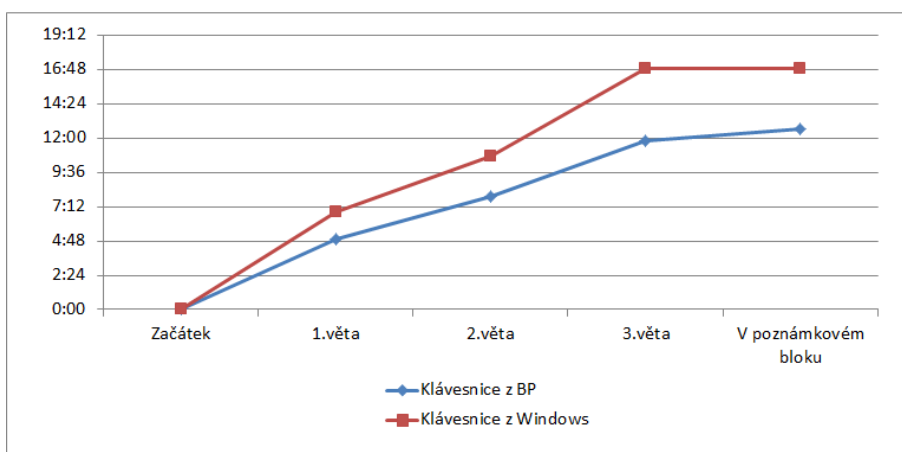
Obrázek 5.3: Průběh psaní jedné věty 2. participantem.



Obrázek 5.4: Porovnání časů participantů při psaní v 4. úkole.



Obrázek 5.5: Porovnání časů účastníků při psaní v 5. úkole.



Obrázek 5.6: Průměrný čas psaní na porovnávaných klávesnicích.

### Rychlost psaní a chybovost

V této části bych se chtěl věnovat zjištěné rychlosti a chybovosti psaní na mnou vytvořené klávesnici. Hned v úvodu bych chtěl uvést, že údaje budou jen orientační. Na zjištění těch reálných by bylo zapotřebí dlouhodobějšího kvantitativního testování. Zde uveřejněné údaje jsou tedy od uživatelů, kteří se setkali s klávesnicí poprvé v rámci mého testování a zároveň na ní měli psát jim do té doby neznámý text. Není tedy výjimkou, že v započítaném čase psaní je i čas strávený čtením. Vzorec pro výpočet WPM, tedy počtu slov napsaných za minutu, je následující:

$$WPM = \frac{CPM}{5}$$

Normou ISO 9241-4 bylo stanoveno, že slovo má v průměru 5 znaků a proto počítáme WPM jako počet znaků za minutu (CPM) dělený pěti. Průměrný čas napsání textu ve čtvrtém úkolu byl 12:41. V tomto úkolu musí účastník stisknout přesně 88 znaků včetně mezer a teček.

Průměrný počet slov napsaný za minutu je tedy přibližně 1,39 WPM. Při stejném testu s uživatelem po pěti hodinách tréninku byl potřebný čas 5:51. Dosáhl tedy 3 WPM. Pro porovnání rychlost na klávesnici z Windows byla 1,04 WPM.

Tabulka 5.1: Hodnoty WPM u participantů na testovaných klávesnicích.

	<b>Klávesnice z BP</b>	<b>Klávesnice z WIN</b>
Participant 1	1,44	1,17
Participant 2	1,73	1,08
Participant 3	1,02	0,86
Participant 4	1,30	0,95
Participant 5	1,71	1,26

Co se týká chybovosti, během úkolů číslo 4 a 5 porovnávajících dvě různé klávesnice byla zaznamenávána každá chyba, kterou uživatel udělal. Ze zjištěných informací vyplývá, že na klávesnici vytvořené pro tuto práci bylo v příslušném úkolu uděláno pěti participanty 9 chyb. Při počtu tří vět tedy v průměru každý udělal 0,6 chyby na jednu napsanou větu. V porovnání při stejném úkolu na klávesnici z Windows bylo pěti participanty uděláno 43 chyb. Tedy 2,86 chyby na jednu větu. Nyní se zaměříme na KSPC, tedy počet úhozů na znak. [17] Je to údaj obsahující informace o chybách, které byly během procesu psaní opraveny.

$$KSPC = \frac{C + INF + IF + F}{C + INF}$$

Pod písmenem C jsou zahrnuty správně napsané znaky, INF jsou znaky neopravené, IF jsou znaky potřebné k opravám jako například backspace a F jsou znaky, které byly při opravě odstraněny. Pro tuto ukázkou nebudeme brát v potaz úspory úhozů na znak získané doplňováním slov ze slovníku. Můj výpočet je o to lehčí, že nedošlo k zanechání žádné chyby v textu. Pro získání výsledků byla dohromady použita data ze všech testů participantů. Výsledné KSPC je tedy přibližně 1,036 úhozů na znak. V porovnání s tím měla klávesnice z Windows v našem testu 1,105 úhozů na znak.

# Kapitola 6

## Závěr

Vytvořená aplikace umožňuje ovládat počítač pohybem hlavy a neverbálním hlasovým vstupem. To vše bez nutnosti použití rukou tak, aby byla uzpůsobena pro lidi s postižením horních končetin. Součástí práce je také prediktivní klávesnice pro český jazyk. Tento celek poskytuje komplexní řešení ovládání počítače, případně je možné samostatně využít každou jeho část zvlášť. Lze ji tak kombinovat s jinými existujícími programy, na které je uživatel zvyklý.

Aplikace byla podrobena testování s uživateli a jeho výsledky poukazují na závislost rozpoznání obličeje na okolních podmínkách. Zejména se jedná o špatné světelné podmínky a nežádoucí barevnost objektů ve scéně. Pokud se jim vyvarujeme, pracuje rozpoznání spolehlivě a lze s ním kurzor ovládat přesně a podle potřeby i rychle. Neverbální hlasový vstup pracuje velmi rychle a s identifikováním gest nemá problémy. Jeho nevýhodou je jen snímání nechtěných hluků z okolí uživatele. Vytvořená prediktivní klávesnice obstála v testu velmi dobře. Její náповěda slov a znaků dokáže uživateli velmi urychlit psaní. Právě zmíněná náповěda dalšího znaku je nápomocná přibližně v 51% případů. To pomáhá v orientaci na jejich novém rozložení, které bylo vytvořeno podle pravděpodobností použití jednotlivých písmen v češtině. Toto rozložení redukuje dráhu, kterou je při psaní nutné s kurzorem urazit. Průměrná rychlost psaní na klávesnici při prvním použití klávesnice se pohybuje okolo průměru 1,39 WPM. Zatím nejvyšší naměřenou rychlostí je hodnota 3 WPM.

V komparativním testu vytvořené prediktivní klávesnice s virtuální klávesnicí z Windows za použití emulace myši z této práce byly zjištěny zajímavé výsledky. Prediktivní klávesnice dosáhla při psaní tří vět o v průměru 24,8% rychlejší čas. Rozdíl jejich rychlostí tedy byl 0,35 WPM. Zároveň při psaní textu na ní vzniklo průměrně 0,6 chyb na jednu větu oproti 2,86 u klávesnice z Windows.

Výsledná aplikace tedy nabízí kvalitní a použitelnou alternativu ke komerčním řešením. I přes to, že lze najít metody ovládání méně závislé na okolním prostředí nebo s ještě větší přesností, mnoha postiženým může pomoci právě mnou prezentované řešení. A to hlavně těm, kteří si nemohou dovolit pořízení přístrojů v ceně desítek tisíc korun.

### Diskuze možného pokračování

Jako vhodné pokračování práce vidím dlouhodobé kvantitativní testování, které by zjistilo limit rychlosti psaní a doporučilo další úpravy uživatelského rozhraní na základě zjištěných

poznatků. Následně by bylo možné upravit layout znaků pomocí optimalizačního algoritmu a zajímavé by také bylo zjišťovat učící křivku klávesnice. Za zamýšlení určitě stojí i myšlenka nahrazení webové kamery zařízením kinect. Dojde tak sice k znatelnému navýšení nákladů na pořízení, avšak díky poměrně přesné hloubkové mapě tohoto zařízení by mělo být snadné a přesné získat s jeho pomocí polohu obličeje. Zároveň bychom se tak mohli zbavit závislosti na vnějších vlivech prostředí. Problémem ovšem zůstává, že toto zařízení je nutné fyzicky připojit k počítači, navíc pomocí speciální redukce na USB. V tomto ohledu je použití integrované kamery v monitoru notebooku výhodnější.

# Literatura

- [1] *Stránky zařízení I4Control* [online]. 2012. [cit. 6. 12. 2012]. Dostupné z: <<http://www.i4control.eu/>>.
- [2] *Seeing With OpenCV, Part 3: Follow that Face!* [online]. 2007. [cit. 10. 12. 2012]. Dostupné z: <[http://www.cognotics.com/opencv/servo\\_2007\\_series/part\\_3/index.html](http://www.cognotics.com/opencv/servo_2007_series/part_3/index.html)>.
- [3] *Stránky programů MyVoice a MyDictate* [online]. 2012. [cit. 18. 12. 2012]. Dostupné z: <<http://www.fugasoft.cz/index.php>>.
- [4] *Stránky programu JetVoice* [online]. 2012. [cit. 18. 12. 2012]. Dostupné z: <<http://www.jrr.cz/jetvoice.php>>.
- [5] *Google books Ngram viewer* [online]. 2012. [cit. 2. 1. 2013]. Dostupné z: <<http://storage.googleapis.com/books/ngrams/books/datasetsv2.html>>.
- [6] *OpenCV wiki* [online]. 2010. [cit. 8. 12. 2012]. Dostupné z: <<http://opencv.willowgarage.com/wiki/>>.
- [7] *Stránky OpenGazeru* [online]. 2009. [cit. 9. 12. 2012]. Dostupné z: <<http://www.inference.phy.cam.ac.uk/opengazer/>>.
- [8] *Frekvence písmen, bigramů, trigramů, délka slov* [online]. 2012. [cit. 14. 12. 2012]. Dostupné z: <[http://nlp.fi.muni.cz/cs/Frekvence\\_pismen\\_bigramu\\_trigramu\\_delka\\_slov](http://nlp.fi.muni.cz/cs/Frekvence_pismen_bigramu_trigramu_delka_slov)>.
- [9] JACOB, R. J. K. What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, s. 11–18, New York, NY, USA, 1990. ACM. Dostupné z: <<http://doi.acm.org/10.1145/97243.97246>>. ISBN 0-201-50932-6.
- [10] MACKAY, D. *Stránky projektu Dasher* [online]. 2011. [cit. 7. 12. 2012]. Dostupné z: <<http://www.inference.phy.cam.ac.uk/dasher/>>.
- [11] Microsoft. *Windows Dev Center* [online]. 2012. [cit. 11. 12. 2012]. Dostupné z: <<http://msdn.microsoft.com/en-us/windows/desktop>>.
- [12] Microsoft. *Jaké funkce usnadnění nabízí systém Windows?* [online]. 2012. [cit. 6. 12. 2012]. Dostupné z: <<http://windows.microsoft.com/cs-CZ/windows7/What-accessibility-features-does-Windows-offer>>.

- [13] O. Poláček and Z. Míkovec. Understanding Formal Description of Pitch-Based Input. *Proceedings of the Third international conference on Human-centred software engineering*. s. 190–197.
- [14] O. Poláček, Z. Míkovec, A. J. Sporka, P. Slavík. Humsher: A Predictive Keyboard Operated by Humming. *In Proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*. , s. 75–82.
- [15] O. Poláček, Z. Míkovec, A. J. Sporka, P. Slavík. New way of vocal interface design: Formal description of non-verbal vocal gestures. *In Proceedings of the 5th Cambridge Workshop on Universal Access and Assistive Technology*. , s. 137–144.
- [16] Občanské sdružení PETIT. *Stránky v sekci Pomůcky* [online]. 2009. [cit. 4. 12. 2012]. Dostupné z: <<http://www.petit-os.cz/>>.
- [17] R. William Soukoreff and I. Scott MacKenzie. Metrics for Text Entry Research: An Evaluation of MSD and KSPC, and a New Unified Error Metric. *Proceedings of the ACM Conference on Human Factors in Computing Systems*. s. 113–120.
- [18] S. Zhai, M. Hunter a B. A. Smith. Performance Optimization of Virtual Keyboards. *HUMAN-COMPUTER INTERACTION*. 2002, 17, s. 89–XXX.
- [19] V. Kalnická, J. Votinský. *Výsledky šetření o zdravotně postižených osobách v České republice za rok 2007* [online]. 2007. [cit. 3. 12. 2012]. Dostupné z: <[http://www.nrzp.cz/dokumenty/Vybrane\\_statisticke\\_udaje\\_OZP\\_2007.pdf](http://www.nrzp.cz/dokumenty/Vybrane_statisticke_udaje_OZP_2007.pdf)>.
- [20] World Health Organisation. *Mezinárodní klasifikace funkčních schopností, disability a zdraví*. 1. U Průhonu 22, Praha 7 : Grada Publishing, 1st edition, 2001.



## Kapitola 7

# Seznam použitých zkratek

**OS** Operační systém

**UI** User Interface - Uživatelské rozhraní

**WPM** Words per minute - Slova za minutu

**CPM** Characters per minute - Znaků za minutu

**KSPC** Keystrokes per character - Počet úhozů na znak

**MIDI** Musical Instrument Digital Interface - Digitální rozhraní pro hudební nástroje

**MSDN** Microsoft Developer Network - Síť vývojců Microsoft

**ASCII** American Standard Code for Information Interchange - Americký standardní kód pro výměnu informací

**URL** Uniform Resource Locator - Jednotný lokátor zdrojů

⋮



## Kapitola 8

# Instalační a uživatelská příručka

Program je určen pro spouštění pod operačním systémem Windows. Byl testován pod verzí Windows 7 64bit a k jeho běhu je navíc potřeba webová kamera a mikrofon. Požadavky na hardware má nízké, na testovacím PC bylo vytížení čtyřjádrového procesoru Intel i7 2.1GHz přibližně 2% a program vyžadoval okolo 250MB paměti RAM. Jeho instalace se provede pouhým zkopírováním celého adresáře aplikace. Dále je nutné nainstalovat aplikaci Java Virtual machine, balíček C++ 2010 redistributable a OpenCV 2.1. OpenCV zároveň vyžaduje software schopný zobrazit okno video výstupu z kamery. Na většině nově zakoupených notebooků je již předinstalovaný program Youcam. Hlavním spouštěcím souborem je `StarPcControl.exe`, který naleznete ve složce PcStarControl.

Každá ze tří hlavních součástí programu je také spustitelná samostatně pomocí příslušných spouštěcích souborů ve složce core. Prediktivní klávesnice může být použita samostatně a vzhledem k tomu, že je psána v jazyce JAVA je multiplatformní, přičemž vyžaduje ke spuštění Java Virtual machine. Lze ji spustit souborem `Keyboard.jar`. Pro Ovládání kurzoru pomocí webové kamery je potřeba nainstalovat balíček Open CV 2.1 a lze jej spustit pomocí `FaceDetService.exe`. Poslední část zajišťující emulaci klikání nonverbálním hlasovým vstupem lze spustit souborem `VoiceKey.exe`. Obě tyto části vyžadují nainstalovaný balíček C++ 2010 redistributable.

### Ovládání

Každá ze tří částí aplikace má také svoje ovládání. U součásti obstarávající ovládání kurzoru lze používat tyto klávesy.

- F11 - Trvale ukončí sledování obličeje a s ním spojené ovládání kurzoru.
- F12 - Tlačítko kalibrace klidového bodu a os pro pohyb kurzoru. Posune klidový bod do současné pozice obličeje a nastaví tak i nové osy, podle kterých se při vychýlení hlavy určuje směr pohybu kurzoru.

U rozpoznání neverbálního hlasového vstupu lze použít následující klávesy.

- F10 - Ztišit mikrofonu
- Šipky nahoru a dolů - Nastavují hranici mezi rozpoznáním vysokého a nízkého tónu.

## Nastavení

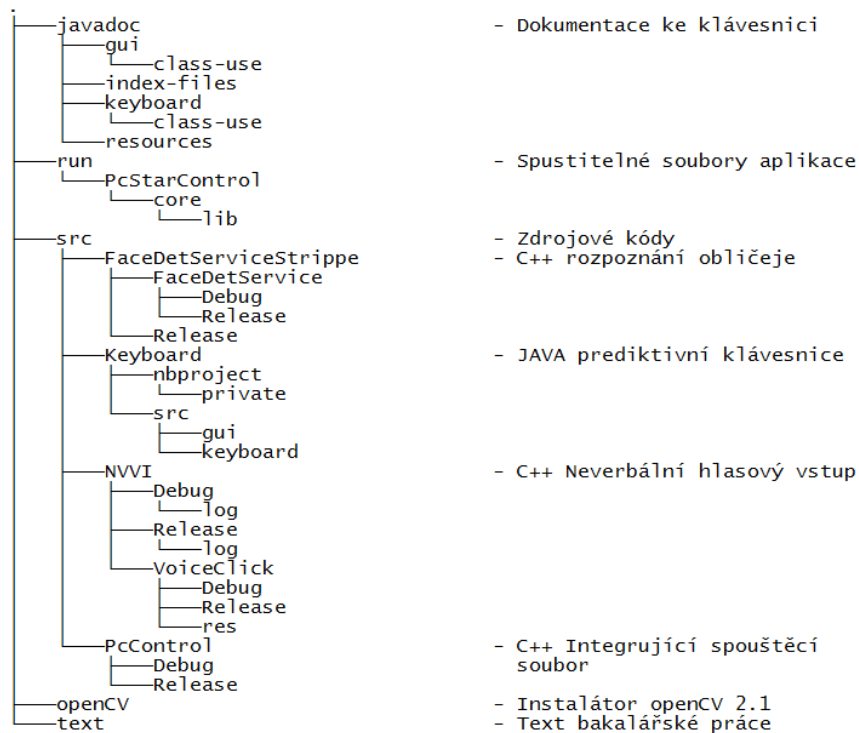
U ovládání kurzoru lze také libovolně měnit nastavení tak, aby byla práce pro uživatele co nejpohodlnější. V kořenové složce aplikace se nachází konfigurační soubor `settings.ini`. Uvnitř něho lze měnit dva parametry.

```
cursorSpeed  
0.3  
quietZone  
15
```

`CursorSpeed` ovlivňuje rychlost pohybu kurzoru a `quietZone` velikost klidové zóny od jejího středu. Velikost zóny je uváděna v počtu bodů ze snímku kamery .

# Kapitola 9

## Obsah příloženého CD



- Dokumentace ke klávesnici
- Spustitelné soubory aplikace
- Zdrojové kódy
- C++ rozpoznání obličeje
- JAVA prediktivní klávesnice
- C++ Neverbální hlasový vstup
- C++ Integrující spouštěcí soubor
- Instalátor openCV 2.1
- Text bakalářské práce

37 directories, 212 files