

Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

Vývoj vstupní metody pro Android využívající Braillovo písmo

Bc. Ondřej Hryzák

Vedoucí: doc. Ing. Daniel Novák, Ph.D.

Obor: Biomedicínská informatika

Studijní program: Biomedicínské inženýrství a informatika

Květen 2018

Poděkování

Rád bych poděkoval vedoucímu mé práce doc. Ing. Danielu Novákovi Ph.D. za to, že mi umožnil pracovat na tomto zajímavém tématu. Dále děkuji všem, kteří mi v průběhu práce pomáhali svými náměty a připomínkami. Děkuji všem lidem, kteří se podíleli v průběhu práce na jejím testování.

Děkuji také své rodině, že jsem se mohl věnovat této práci.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 25. května 2018

Abstrakt

Tato práce se zabývá možnostmi vstupu Braillova písma na mobilních zařízeních se systémem Android. V úvodní části jsou popsány a porovnány vybrané systémy pro Braillov vstup, a také jednotlivé typy Braillova písma.

V další části práce je navržena samotná implementace Braillovy klávesnice pro Android. Důležitou součástí je návrh uživatelského rozhraní klávesnice. Zde je kladen velký důraz na co nejlepší přizpůsobení ovládání nevidomým uživatelům tak, aby klávesnice byla co nejjednodušeji ovladatelná a zároveň bylo dosaženo vysoké přesnosti při psaní. K tomu slouží například úprava polohy jednotlivých kláves na základě historie uživatelských stisků. To je velice důležité vzhledem k absenci hmatové zpětné vazby. V práci je navrženo více metod, pomocí kterých lze pozici jednotlivých kláves upravovat. I přes tuto korekci může docházet k chybám (gramatická chyba nebo překlep). Proto se práce také zabývá možnostmi automatické korekce již napsaných slov. K dalšímu usnadnění a zrychlení psaní je část práce věnována možnostem doplňování rozepsaných slov.

V závěru práce je provedeno testování klávesnic na nevidomých uživateli. V testech jsou zjišťovány chybovosti při psaní, doba psaní a jaké typy chyb vznikají a co je jejich příčinou.

Klíčová slova: Android, Braille, klávesnice, asistivní technologie

Vedoucí: doc. Ing. Daniel Novák, Ph.D.

Abstract

This thesis deal with the possibilities of entering Braille on Android mobile devices. The first part describes and compares selected systems for Braille input as well as individual types of Braille.

In the next part of the thesis, the implementation of the Braille keyboard for Android is designed. The important part is the design of a keyboard user interface. The emphasis is on optimizing the controls for blind users so that the keyboard usage is as easy as possible while maintaining high accuracy when writing. Example of this would be adjusting the position of individual keys based on the history of users presses. This is very important due to the absence of tactile feedback. This thesis proposes several methods to modify the position of the individual keys. Despite this correction, gramatical or typing arrors may occure. Because of that, the thesis also deals with the possibilities of automatic correction of already written words. To further simplify and accelerate writing, part of the thesis is devoted to the possibilities of completing the words. Last part of the thesis focuses on tests/testing that was performed by blind users. Writing errors, typing times, and what types of errors are occurring and what causes them are identified in the tests.

Keywords: Android, Braille, keyboard, assistive technology

Title translation: Development of an Android Input Method Using Braille

Obsah

1 Úvod	1
---------------	----------

Část I Řešení vstupu Braillova písma

2 Historie Braillova písma	5
-----------------------------------	----------

2.1 Vývoj vstupu Braillova písma ...	6
--------------------------------------	---

2.2 Vývoj a typy Braillova písma	8
---------------------------------------	---

3 Současná řešení Braillova vstupu na mobilních zařízeních	11
---	-----------

3.1 Softwarová řešení	11
-----------------------------	----

3.1.1 BrailleTouch	12
--------------------------	----

3.1.2 SoftBraille Keyboard	13
----------------------------------	----

3.1.3 SwiftBraille Keyboard	14
-----------------------------------	----

3.1.4 iOS Braille Screen Input	15
-------------------------------------	----

3.2 Hardwarová řešení	16
-----------------------------	----

3.2.1 HoliBraille	16
-------------------------	----

3.2.2 Bluetooth klávesnice	17
----------------------------------	----

3.3 Převod mezi Braillovými znaky a abecedou	18
--	----

Část II Navržené řešení a implementace

4 Návrh řešení	23
-----------------------	-----------

4.1 Klíčové požadavky na klávesnici	24
-------------------------------------	----

5 Návrh uživatelského rozhraní	27
---------------------------------------	-----------

5.1 Rozvržení klávesnice	27
--------------------------------	----

5.1.1 Převod Braillova písma na běžný text	28
--	----

5.2 Ovládání	29
--------------------	----

5.2.1 Psaní textu	29
-------------------------	----

5.2.2 Gesta	30
-------------------	----

5.2.3 Kontextové menu	34
-----------------------------	----

5.2.4 Úprava textu	35
--------------------------	----

6 Dynamická pozice kláves	37
----------------------------------	-----------

6.1 Klouzavý průměr	38
---------------------------	----

6.2 Kálmánův filtr	38
--------------------------	----

7 Predikce psaných slov	43
--------------------------------	-----------

7.1 Český národní korpus	44
--------------------------------	----

7.2 Dokončování slov pomocí prefixového stromu	44
7.2.1 Implementace stromu	45
7.2.2 Prohledávání stromu	45
7.3 Autokorekce slov pomocí BK-stromu	48
7.3.1 Implementace BK-stromu ...	50
7.3.2 Průběh autokorekce slov	52
7.4 Ukládání vytvořených stromů ..	53

Část III Uživatelské testování

7.5 Úvodní dotazování	57
7.6 Testování aplikace	58
7.7 Zpětná vazba	59
7.8 Vyhodnocení chybovosti klávesnice	60
7.8.1 Total Error Rate	60
7.8.2 MSD Error Rate (minimum string distance)	60
7.8.3 KSPC (Key Strokes per Character)	61

7.8.4 WPM (Words per Minute) ..	61
7.9 Uživatelské testování	62
7.9.1 Uživatel K	62
7.9.2 Uživatel R	63
7.9.3 Uživatel B	64
7.10 Vyhodnocení testování	66

Část IV Závěr

Přílohy

A Výsledky testování	75
A.1 Výsledky úvodního dotazování .	75
A.2 Výsledky závěrečného dotazování	75
A.2.1 Uživatel K	75
A.2.2 Uživatel R	76
A.2.3 Uživatel B	77
A.3 Testovací text	77
A.3.1 Text zapsaný na iPhone pomocí iOS Braille Screen Input .	78

A.3.2 Text zapsaný na Android pomocí navrženého řešení	78
B Obsah CD	79
C Literatura	81
D Zadání práce	85

Obrázky

2.1 Porovnání písmen 'a' a 'z' v různých hmatových písmech	6	3.7 Architektura modulů překladu [7]	18
2.2 Pichtův stroj [18]	7	4.1 Problémové hrany mezi klávesami	24
2.3 Přiřazení bodů Braillova písma na klávesnici	8	5.1 Rozvržení počátečních bodů kláves	28
2.4 Rozměry Braillova znaku	9	5.2 Hodnota jednotlivých kláves pro sestavení unikátního kódu znaku abecedy	29
2.5 Tabulka české verze Braillovy abecedy [17].	9	5.3 Vložení mezery / potvrzení úpravy	30
2.6 Porovnání zápisu slova "wedding" v plnopisu a zkratkopisu.	10	5.4 Mazání znaku / zrušení úpravy . .	31
3.1 Náhled rozvržení klávesnice BrailleTouch	13	5.5 Čtení celého textu	32
3.2 Náhled rozvržení klávesnice SoftBraille [18]	13	5.6 Posun na předchozí slovo	32
3.3 Náhled rozvržení klávesnice SwiftBraille	14	5.7 Posun na další slovo.	32
3.4 Náhled klávesnice iOS Braille Screen Input	15	5.8 Doplnění slova pro prefix nebo autocorrect	33
3.5 Zařízení HoliBraille s dotykovým telefonem	17	5.9 Smazání celého textu.	34
3.6 Bluetooth klávesnice BraillePen Slim [9].	17	7.1 Prefixový strom s četnostní slova c	46
		7.2 Průběh doplnění slova	49
		7.3 Ukázka BK-stromu	51
		7.4 Průběh opravy slova	54

Tabulky

1.1 Počty nevidomých lidí na světě ..	2
7.1 Prefixy s maximálním počtem procházených uzlů	48
7.2 Psaní písmen	63
7.3 Psaní slov	64
A.1 Úvodní dotazování	75

Kapitola 1

Úvod

V současné době žije na celém světě podle odhadu WHO z roku 2015 přibližně 258 miliónů lidí se zrakovým postižením, z toho je téměř 40 miliónů nevidomých [15]. Počet nevidomých tak stále narůstá. Odhad v roce 1990 byl 30,6 miliónů nevidomých lidí [4]. V roce 2050 by již mělo být na světě 114,6 miliónů nevidomých [5]. To je způsobeno zvyšováním počtu obyvatel, stárnutím populace a dalšími sociálněekonomickými aspekty, jako například dostupností lékařské péče. Velká část z těchto nevidomých navíc žije v rozvojových zemích, kde mají jen velmi omezený přístup ke speciálním pomůckám, které by jim usnadnili práci, komunikaci s ostatními lidmi, a tím i zlepšili jejich život.

Zároveň však v současnosti dochází k ohromnému rozvoji chytrých mobilních telefonů, kdy se tyto zařízení stávají i levnější a tím pádem mají možnost dostat se k daleko větší skupině lidí. V současné době je na světě 5 miliard uživatelů mobilního telefonu. [?] Mobilní telefony, ať chceme nebo ne, ovlivňují naše životy čím dál tím více. Využíváme je jako jeden z hlavních prostředků pro komunikaci, práci, ale i zábavu. Pro mnoho lidí by bylo nepředstavitelné trávit čas bez možnosti využít chytrý mobilní telefon. V dnešním světě totiž hraje elektronická komunikace v kontaktu mezi lidmi stále důležitější roli. Tento rozvoj a rozšiřování chytrých mobilních telefonů ale způsobuje nevidomým řadu problémů. Telefony jsou většinou velmi špatně přizpůsobené pro jejich využívání právě nevidomými. Velká část nevidomých lidí proto stále používá staré tlačítkové telefony než aby začali používat dotykové telefony. Staré telefony pro ně mají výhodu v tom, že obsahují právě mechanická tlačítka, díky nimž mají hmatovou zpětnou vazbu o každém svém dotyku. Nevýhodou je, že tyto telefony jsou použitelné pouze pro základní funkce jako telefonování nebo posílání SMS. Tyto telefony často ani nepodporují žádným způsobem zvukovou zpětnou vazbu, uživatel si tak pouze mechanicky

Věk	Populace	Nevidomí	Slabozrací	Zrakově postižení
0-14	1 848 500 000	1 421 000	17 518 000	18 939 000
15-49	3 548 200 000	5 784 000	74 463 000	80 248 000
50+	1 340 800 000	32 160 000	154 043 000	186 203 000
Suma	6 737 500 000	39 365 000	246 024 000	285 389 000

Tabulka 1.1: Počty nevidomých lidí na světě

pamatuje postup například pro vytočení čísla.

Jedna z otázek může znít, proč by měli nevidomí používat dotykové telefony? Bohužel jsou k tomu nuceni popsánymi okolnostmi. Určitě by pro nevidomé bylo příjemnější využívat tlačítkové telefony, jako to dělali doposud. Tím by ale přicházeli o možnost využívat možnosti, které poskytuje právě chytrý telefon. Ty totiž téměř neexistují v tlačítkové verzi. Při vývoji nových telefonů by tak mělo být myšleno na všechny skupiny uživatelů, kteří ho mohou využívat a přizpůsobit ho. To se v současné době úplně neděje. Chytré telefony nabízejí možnosti, které by nevidomým mohli výrazně ulehčit život

Člověka také může napadnout otázka, zda vůbec potřebují nevidomí uživatelé ovládat chytrý telefon pomocí klávesnice. V dnešní době, kdy každý chytrý telefon má v sobě zabudované hlasové ovládání, by mohli psát texty pouze pomocí hlasového diktování. A to jak v případě telefonů s Androidem, tak i iOS.

Hlasové diktování je sice pohodlné, ale není úplně vždy použitelné. Může se jednat o rušné prostředí, kde je tolik šumu, že ve výsledném textu vznikne nepřiměřené množství chyb. Nevidomý uživatel by je pak měl problém opravit a spíše by ho to zdržovalo. Mohou také nastat situace, kdy není vhodné text diktovat. Například při jízdě hromadnou dopravou, kdy nechcete text diktovat. To může nastat také při jiných hromadných akcích, přednáškách a podobně. Navíc diktování textu je trochu jiné, než když člověk text píše. Je to podobné jako se čtenými knížkami. Nevidomí uživatelé si v některých případech knížku raději čtou, než poslouchají načtenou. Stejně tak, jako vidomí uživatelé téměř výhradně využívají klasickou klávesnici, i když by mohli také využít diktování.

Cílem práce tak je navrhnout takové řešení pro Braillův vstup, aby by bylo co nejvíce přívětivé pro nevidomé uživatele a doplnilo tak stávající možnosti dostupné pro ovládání telefonu s operačním systémem Android. Tím by se rozšířili možnosti pro používání těchto telefonů nevidomými uživateli. V současné době je možná stále dost nevidomých uživatelů, kteří nechtějí chytré telefony. S tím, jak ale bude vyrůstat digitální generace, která je po celou dobu v kontaktu s chytrou elektronikou, získají řešení pro nevidomé na mobilních telefonech daleko větší uplatnění.

Část I

Řešení vstupu Braillova písma

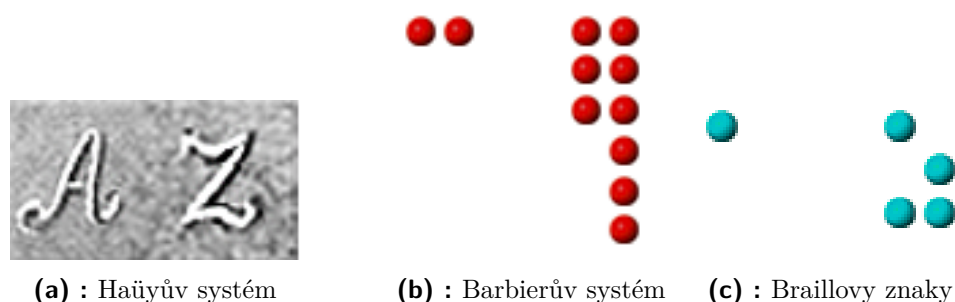
Kapitola 2

Historie Braillova písma

Historie Braillova písma sahá až do první půlky 19. století [16]. Postaral se o to Francouz Louis Braille, který během svého dětství vlivem zranění a následné infekce oslepl. Začal docházet na Institut des Jeunes Aveugles, což byl institut pro slepou mládež. Braille se snažil přijít na způsob, jak by nevidomý lidé mohli psát text. Do té doby se pro nevidomé využíval systém Valentina Haiüye, který spočíval ve vylisování šablon písmen standardní abecedy do tvrdého papíru. Tím vznikla na papíru vystouplá struktura textu, která byla hmatem rozeznatelná. Čtení však šlo pomalu, protože jednotlivá písmena jsou v některých případech těžko rozeznatelná, a navíc nebylo možné ručně psát text. Sice stále probíhaly pokusy o to, jak jednotlivé znaky abecedy co nejvíce zjednodušit a odlišit, aby šly rychleji číst, ale k většímu úspěchu to nevedlo.

Louis Braille se proto snažil přijít na způsob, jak to změnit a umožnit nevidomým číst i zapisovat text. Inspirací mu nakonec bylo vojenské noční písmo vynalezené Charlesem Barbierem de la Serre. To bylo využíváno původně v armádě, aby vojáci mohli komunikovat během noci bez nutnosti svítit si při práci s textem a odhalit tak například svoji pozici. Písmo se skládá z dvanáctibodových znaků (2 sloupce po šesti bodech) vytlačených do papíru. Počet bodů v každém sloupci určuje pozici znaku v Polybiově čtverci. Ten tak obsahuje maximálně 36 pozic s konkrétními znaky nebo skupinou znaků. Charles Babier zvolil četné francouzské skupiny písmen, ze kterých šla sestavit téměř všechna slova.

Tato metoda nočního písma inspirovala právě Louise Braille, který se s metodou seznámil při návštěvě vojenského důstojníka na institutu, kde právě studoval. Barbierova metoda byla příliš složitá a specifická, než aby ji bylo možné stejně využívat i pro běžné civilní účely. Navíc šest bodů ve sloupci



(a) : Haüyův systém (b) : Barbierův systém (c) : Braillovy znaky

Obrazek 2.1: Porovnání písmen 'a' a 'z' v různých hmatových písmech

neumožňovalo pohodlné čtení, protože znak byl delší než bříško prstu, ten se tak po znaku musel posouvat. Proto Braille změnil systém, zredukoval počet bodů na 6 (dva sloupce po třech bodech), což už umožňovalo obsáhnout celý znak bříškem prstu. Snažil se také co nejvíce využít všech kombinací. Pro šest bodů tak existuje 2^6 kombinací bodů pro zakódování znaku. Navíc v první verzi používal kromě bodů i pomlčku, která spojovala dva body na jednom řádku. Svoji první verzi dokončil již v 15 letech v roce 1824 a finální verzi vydal v roce 1829. Písmo se ale prosazovalo pomalu, důvodem bylo i to, že písmo bylo pro běžného člověka nečitelné. Souběžně tak stále pokračovaly pokusy o vytvoření slepeckého písma využívajícího latinku.

Braille však také stále pracoval na vylepšení svého písma. Doplnil například možnost zapisovat hudební noty pomocí tohoto písma. Druhou verzi písma vydal v roce 1837. Tam již vypustil pomlčky, protože se ukázalo, že jsou oproti samostatným bodům špatně čitelné a rozeznatelné hmatem. Vznikla tak verze, jejíž základ je využíván dodnes. Mezi nevidomými začalo písmo získávat brzo podporu, ale stále

2.1 Vývoj vstupu Braillova písma

Od počátku užívání Braillova písma se lidé snažili o co nejjednodušší možnost jeho zápisu. Louise Braille zpočátku pro zápis využíval pouze hrot šídla, kterým si paradoxně v dětství slepotu způsobil. Poté se využívá vymačkávání bodů skrz tabulku s body. Modifikací vzniká na konci 19. století v pražském Klarově ústavu [16] například tzv. pražská tabulka. Ta se skládá ze dvou kovových listů, mezi něž se vkládá papír. V horním plátu jsou obdélníky pro jednotlivé znaky s šesti polokulatými průřezy, skrz které se vytlačují jednotlivé body do papíru. Ve spodním plátu jsou vylisovány pro každý obdélník všechny body. To zapříčiní to, že všechny promáčké body mají stejnou hloubku a tvar. Zásadním problémem tohoto typu psaní bylo, že se jednotlivé znaky musely psát zrcadlově a směrem zprava doleva. Teprve po vyjmutí papíru z

plátů bylo možné text přečíst. U každého znaku se navíc musely jednotlivé body vymačkávat jednotlivě, což zpomalovalo psaní. Další nevýhodou je to, že psaný text nejde zároveň číst a tak napsaný text například kontrolovat. Přesto se pražská tabulka, nebo její modifikace, s oblibou pro svou jednoduchost a přesnost využívá dodnes.

Na přelomu 19. a 20. století začali pokusy o sestrojení mechanického psacího Braillova stroje. Jedním ze strojů, který zaznamenal větší úspěch a je stále využíván, je stroj od německého tyflopeda Oskara Pichta. Ten ho sestrojil již v roce 1899, konstrukce je obdobná jako u klasických psacích strojů. Skládá se z pohyblivého válce s papírem a statické razící hlavy. Dále pro samotné psaní obsahuje šest kláves pro jednotlivé body znaku a jednu klávesu jako mezerník. Výhodou psaní na stroji je značné urychlení psaní oproti tabulkovému psaní. Každý znak se napíše najednou kombinací stisků jednotlivých kláves. Navíc se na rozdíl od psaní pomocí tabulek vytváří pozitivní text, takže ho lze zároveň při psaní i číst.

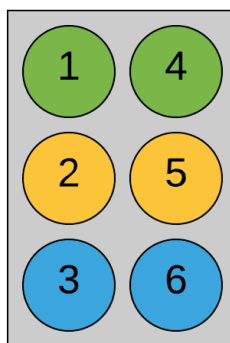


Obrázek 2.2: Pichtův stroj [18]

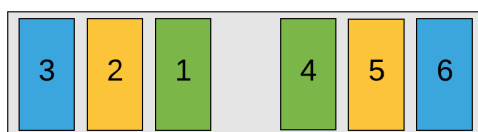
Pichtův stroj se stal rázem oblíbený. Kromě rychlosti psaní to bylo způsobeno i jeho kompaktními rozměry. V průběhu 20. století docházelo k jeho dalšímu vývoji, například vytvoření elektronických verzí. Základní princip a rozložení kláves zůstalo stejné do dnešní době pro téměř všechny vstupní zařízení pro Braillovo písmo.

Vznikaly i další stroje, které se lišily konstrukčně. Za zmínku stojí například Perkinsův stroj. Ten se liší hlavně konstrukčně, namísto pohyblivého se válce obsahuje pohyblivou se razící hlavu. Kromě toho má navíc klávesy pro nový řádek a backspace pro opravování napsaných znaků.

U všech těchto mechanických strojů se využívalo stejné rozložení základních kláves. Rozložení vychází z principu, kdy bod 1 Braillova znaku je umístěn na vnitřní straně levé trojice kláves a na vnitřní straně pravé trojice je bod 4.



(a) : Braillov znak



(b) : Rozložení klávesnice

Obrázek 2.3: Přiřazení bodů Braillova písma na klávesnici

Přiřazení jednotlivých bodů na klávesnici je zobrazeno na obrázku 2.3. Stejné rozložení se udrželo u většiny vstupních zařízení, ať už Braillových klávesnic mechanických nebo softwarových na mobilních zařízeních.

2.2 Vývoj a typy Braillova písma

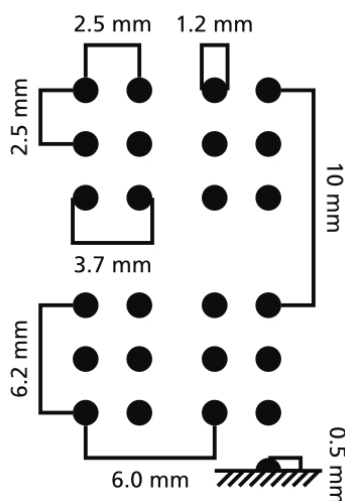
Na základě typu Braillova písma vydaného v roce 1837 začaly vznikat postupem času další rozšíření. Pro každou oblast, kde bylo potřebné, aby nevidomí mohli zapisovat nějaké informace, vznikla další rozšířená znaková sada. Například vznikla verze pro zapisování hudebních not, rozšířená matematická verze se speciálními znaky a podobně.

Kromě toho je často nutné vytvořit vlastní úpravu pro odlišné jazyky a jejich znakové sady. Ty totiž mohou obsahovat rozdílné znaky, které je třeba zachytit, například znaky s interpunkcí. Základní sada 26 znaků je v celé Evropě téměř shodná. Využívá se pro psaní složené ze znaků shodně s běžným textem, to znamená, že každé písmeno textu se zapíše pomocí jednoho znaku. Tento způsob se v anglické verzi Braillova písma nazývá Grade 1.

Bohužel při tomto způsobu psaní, kdy jeden znak textu odpovídá jednomu znaku Braillova písma, jsou značné typografické problémy. Všechny znaky mají jednotnou velikost, takže zápis je oproti běžnému textu prostorově náročnější. Na jeden řádek se vejde přibližně 30 znaků. Znak má navíc na

výšku, včetně oddělovací mezery, rozměr přibližně 10 mm, jak je zobrazeno na obrázku 2.4.

Jedním z problémů Braillova písma je to, že neexistuje žádná centrální



Obrázek 2.4: Rozměry Braillova znaku

autorita pro definici základních principů písma atablek pro znaky, které by jednotlivá národní písma sjednocovala. Neexistuje tak jednotná velikost ani tabulka základních znaků. Bohužel ani v České republice neexistuje oficiálně žádná autorita, která by určovala pravidla pro Braillovo písmo. V této práci vycházím z tabulek spravovaných sdružením SONS (Sjednocená organizace nevidomých a slabozrakých České republiky) [17], kde lze nalézt tabulky šestibodého Braillova písma pro plnopis (Grade1) se znaky české abecedy.

a (1)	⠁	g (1245)	⠒	ó (246)	⠏	ů (23456)	⠛
á (16)	⠁⠠	h (125)	⠓	p (1234)	⠏	v (1236)	⠜
b (12)	⠃	i (24)	⠒	q (12345)	⠏⠞	w (12356)	⠜⠞
c (14)	⠉	í (34)	⠒⠠	r (1235)	⠓	x (1346)	⠜⠞
č (146)	⠉⠠	j (245)	⠒⠞	ř (2456)	⠓⠞	y (13456)	⠜⠞⠞
d (145)	⠉	k (13)	⠓	s (234)	⠓	ý (12346)	⠜⠞⠞
ḍ (1456)	⠉⠞	l (123)	⠓	š (156)	⠓⠞	z (1356)	⠜⠞⠞
e (15)	⠑	m (134)	⠓	t (2345)	⠓⠞	ž (2346)	⠜⠞⠞
é (345)	⠑⠠	n (1345)	⠓⠞	ť (1256)	⠓⠞⠞		
ě (126)	⠑⠞	ň (1246)	⠓⠞⠞	u (136)	⠓⠞⠞		
f (124)	⠑	o (135)	⠓	ú (346)	⠓⠞⠞		

Obrázek 2.5: Tabulka české verze Braillovy abecedy [17]

Grade1 - plnopis. Plnopis představuje formu zápisu znaků Braillova písma, kdy vždy jeden znak Braillova písma představuje jeden znak z abecedy. Tento způsob zápisu je nejjednodušší na naučení. Vychází z původního Braillova návrhu z roku 1837. Protože šestibodý systém umožňuje zakódovat pouze 63 znaků, je nutné využívat i prefixy pro speciální znaky (velká písmena, čísla, řecká písmena). Tabulka základních písmen je téměř pro všechny jazyky shodná, liší se většinou pouze ve speciálních znacích s interpunkcí.

Grade2 - zkratkopis. Zkratkopis se vyvinul z několika důvodů. Jedním z nich byla potřeba úspory místa při psaní, kdy psaní plnopisem využívá velkou plochu papíru. Kromě toho zkratkopis zrychluje psaní. Je založen na tom, že jeden znak Braillovy abecedy kóduje více znaků abecedy. Z toho vychází, že je velmi individuální pro každý jazyk, protože je nutné vytvořit skupiny písmen abecedy, které se v daném jazyku často vyskytují spolu, a každé skupině přidělit Braillov znak. Tento způsob se využívá hlavně v angličtině nebo němčině, kde převažuje nad plnopisem.



(a) : Plnopis - UK Grade1
w-e-d-d-i-n-g



(b) : Zkratkopis - UK Grade2
w-ed-d-ing

Obrázek 2.6: Porovnání zápisu slova "wedding" v plnopisu a zkratkopisu

Osmibodové Braillovo písmo. Postupem času se vyvinulo rozšíření klasického šestibodového písma o dva body na písmo osmibodové [8]. Tím je získána možnost využít 255 kombinací bodů (prázdné body jsou pro mezeru). Základní znaky zůstávají stejné jako v plnopisu. Hlavní rozdíl nastává v tom, že není potřeba využívat prefixy pro speciální znaky, jako jsou například velká písmena nebo číslovky. Prefix je obsažen rovnou v sedmém a osmém bodu. Tím opět dochází ke zrychlení psaní i úspoře místa. Využívá se ale hlavně na elektronických braillových řádcích připojených k počítači.

Kapitola 3

Současná řešení Braillova vstupu na mobilních zařízeních

V této části práce je proveden rozbor toho, jaká existují dostupná řešení pro Braillov vstup na mobilní telefony, ať už s operačním systémem Android nebo iOS. Při hledání již realizovaných nebo navrhovaných řešení bylo vycházeno ze zdrojů, které se tímto problémem již zabývali, nebo jsem se snažil nalézt realizovaná aktuálně dostupná řešení pro jednotlivé chytré telefony (Android, iOS) přímo v obchodech s aplikacemi (Google Play nebo App Store).

U jednotlivých nalezených řešení byla provedena analýza toho, jaké silné a slabé stránky dané řešení má, hlavně z pohledu uživatele.

Obecně lze možnosti vstupu rozdělit do dvou hlavních skupin v závislosti na přístupu. Jsou to hardwarová řešení, kdy se k mobilnímu telefonu připojí externí Braillova klávesnice, například přes Bluetooth. Druhou skupinou jsou softwarová řešení, kdy se vstupem Braillova písma stává samotný dotykový displej mobilního telefonu. Každý z těchto přístupů má svoje výhody i nevýhody. Jaké to jsou konkrétně jsem více rozebral v konkrétních kapitolách 3.1 a 3.2.

3.1 Softwarová řešení

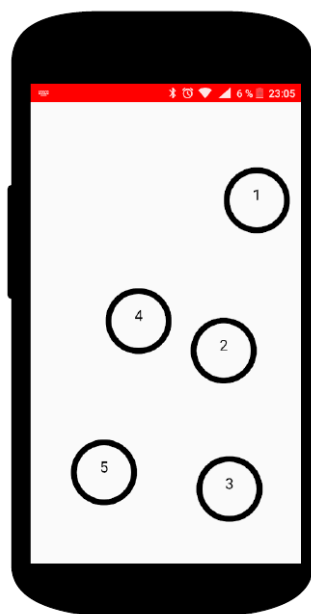
Softwarová řešení vstupu Braillova písma představují nejjednodušší způsob, jak lze na mobilním telefonu Braillovým písmem psát. Stačí si pouze na mobilní telefon stáhnout a nainstalovat klávesnici, kdy je na výběr z množství



Obrázek 3.1: Náhled rozvržení klávesnice BrailleTouch

■ 3.1.2 SoftBraille Keyboard

SoftBraille klávesnice je už založena na dynamické poloze kláves. Jednotlivé bloky kláves 1, 2, 3 pro levou a 4, 5, 6 pro pravou ruku jdou na začátku zkalibrovat. Tím dojde k vytvoření rozmístění jednotlivých kláves v obou blocích přímo na rozložení uživatelových prstů. Ty pak reagují na jednotlivé dotyky a celý blok upraví svoji pozici. Jak je vidět na obrázku 3.2, blok kláves se může dostat až za hranici displeje. V tu chvíli musí dojít k nové kalibraci, protože jinak není možné zapisovat všechny znaky.



Obrázek 3.2: Náhled rozvržení klávesnice SoftBraille [18]

Výhody. Mezi výhody této klávesnice lze zařadit možnost psaní na šířku i výšku displeje. Možnost psaní pomocí jednoho tahu zase zpřístupňuje možnost vytvářet text i uživatelům, kteří z nějakého důvodu nejsou schopni

Nevýhody. Nevýhodou jsou malá tlačítka klávesnice. Nevidomý, který má problém přesně určit konkrétní místo na displeji, může mít problém při klasickém způsobu psaní s přesností.

Další problém je při samotném psaní tahem. Pokud chce uživatel stisknout znak K (klávesa 1 a 3), musí tahem označit tyto dvě klávesy a přitom se vyhnout klávese 2. To se jeví jako docela náročný úkol pro nevidomého uživatele.

Problémem také je pevně daná pozice jednotlivých kláves, kdy není možné jejich přizpůsobení konkrétnímu uživateli.

Nepřesnosti může způsobovat i otočené sloupce bodů na displeji. Ty jsou opačně, než je běžným zvykem. Pod levým ukazováčkem je při psaní klávesa 4 místo klávesy 2.

3.1.4 iOS Braille Screen Input

Apple jako výrobce mobilního operačního systému iOS na rozdíl od Androidu ve svém systému poskytuje komplexní podporu pro nevidomé, zahrnující i integrovanou klávesnici na Braillův vstup. Na obrázku je zobrazen náhled klávesnice. Lze využívat variantu s šesti body, tak i osmibodovou klávesnicí. Ta je ale z několika důvodů dostupná pouze na zařízeních s větším displejem (iPad). Je to nejspíše proto, že na malém displeji by docházelo k nepřesnostem. Prsty stisklé blízko u sebe jsou špatně detekovatelné od sebe. Dalším důvodem je, že na malém displeji není ani dostatek místa na pohodlné psaní osmi prsty.



Obrázek 3.4: Náhled klávesnice iOS Braille Screen Input



Obrázek 3.5: Zařízení HoliBraille s dotykovým telefonem

■ 3.2.2 Bluetooth klávesnice

Kategorie Bluetooth klávesnic přináší uživatelům daleko lepší komfort při psaní než softwarové klávesnice. To je způsobeno hmatovou zpětnou vazbou, která u softwarových řešení na dotykových displejích chybí. Bluetooth klávesnice vychází svojí konstrukcí z mechanických Braillových strojů. Obsahují šest kláves pro jednotlivé body a speciální klávesy pro další funkce (mezera, zpět, mazání) v závislosti na konkrétní variantě.



Obrázek 3.6: Bluetooth klávesnice BraillePen Slim [9]

Mezi nevýhody hardwarových klávesnic lze zařadit jejich cenu. Zatímco některé softwarové klávesnice lze získat jako aplikaci zdarma, hardwarové klávesnice jsou větší investicí, ta může několikrát převyšovat cenu samotného mobilního telefonu. Tím se mohou stát méně dostupné pro některé nevidomé uživatele, obzvláště v rozvojových zemích. Další nevýhodou je rozměr. Ten je sice relativně malý, ale vyžaduje to stále nošení dalšího zařízení kromě samotného mobilního telefonu. Psát pak není úplně snadné kdekoliv, protože musí být někde umístěn telefon a zvláště klávesnice.

Braillova písma, protože pravidla jsou často nejednoznačná (slovo lze například rozdělit na skupiny znaků více způsoby, kterou z nich použít?).



Část II

Navržené řešení a implementace

Kapitola 4

Návrh řešení

V této části se práce zabývá návrhem řešení klávesnice pro Braillov vstup. Na základě provedené analýzy již existujících řešení [19] se jako nejlepší řešení jeví využití rozvržení se všemi šesti body na displeji najednou. Tento způsob zápisu dosahuje nejvyšší rychlosti psaní, ovšem velké chybovosti (BrailleTouch) [19]. Cílem tak je pokusit se navrhnout řešení takové, aby byla chybovost psaní minimalizována, ale zachována rychlost.

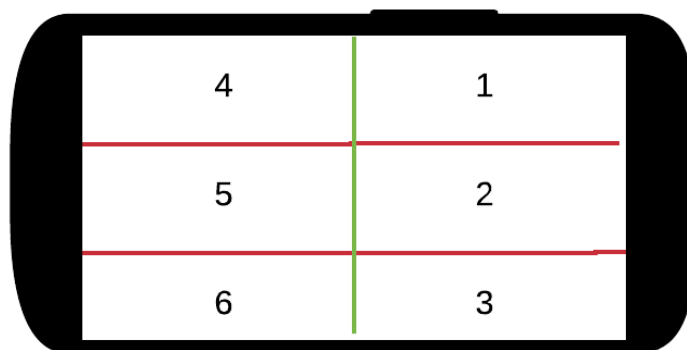
Klávesnice by také měla být co nejvíce funkčně přizpůsobena na požadavky samotných nevidomých uživatelů. Proto bylo ještě před samotným návrhem provedeno úvodní testování vybraných aplikací (iOS Braille Screen Input, SwiftBraille), aby bylo možné lépe identifikovat jednotlivé problémy, se kterými se musejí nevidomí uživatelé potýkat při psaní na dotykovém displeji mobilního telefonu. Během tohoto testování se také

Návrh a implementace řešení je možné rozdělit na tyto jednotlivé části:

- Návrh a implementace uživatelského rozhraní
- Návrh a implementace dynamické pozice kláves
- Návrh a implementace predikce psaných slov

4.1 Klíčové požadavky na klávesnici

Při návrhu klávesnice pro nevidomé nelze vycházet pouze z předpokladů, které má člověk s nepoškozeným zrakem. Na úvod tak bylo nutné zjistit požadavky na jednotlivé klíčové prvky klávesnice od samotných nevidomých uživatelů. To proběhlo ve spolupráci s centrem SONS. Cílem bylo získat informace o způsobech užívání klávesnice, potřebných funkcionalitách pro psaní, které by měla klávesnice obsahovat. Součástí bylo i uživatelské testování různých druhů klávesnice pro zjištění stylu psaní na tomto typu klávesnic a toho, co může způsobovat problémy při psaní.



Obrázek 4.1: Problémové hrany mezi klávesami

Jak je zobrazeno na obrázku 4.1, bylo zjištěno, že největší problém při psaní způsobují horizontální hrany mezi klávesami, což se projevilo právě během úvodního testování. Je to důsledek toho, že nevidomý uživatel neúmyslně mění pozici prstů na displeji během psaní. Často se pak dostane prstem až na rozhraní jednotlivých vertikálních sektorů a vznikají tím nepřesnosti při psaní, kdy je rozeznán jiný znak, než jaký zamýšlel vložit uživatel. Tento druh chyb je zastoupen i v analýze chybovosti klávesnic v článku [19].

Z tohoto úvodního otestování některých dostupných řešení bylo možné identifikovat klíčové požadavky, které by měla navržená klávesnice odrážet. Snahou je vyvarovat se problémů, které byly nalezeny u testovaných řešení a implementovat co nejvhodněji požadavky od nevidomých uživatelů.

Klíčové vlastnosti, které by měla klávesnice splňovat:

- **Dynamická pozice kláves** Na základě stisků uživatele přizpůsobovat dynamicky v průběhu psaní jednotlivé pozice kláves.
- **Maximální využití plochy displeje** Navrhnout ovládací prvky tak,

aby nevyužívaly displej, a ten byl použit pouze pro klávesy Braillova bodu.

- **Intuitivní ovládání** Přizpůsobit ovládání nevidomým tak, aby bylo jednoduše zapamatovatelné (gesta i menu).
- **Funkce k urychlení psaní** Na základě zadaných znaků provádět doplnění slov nebo opravovat automaticky překlapy. Navrhnout gesta tak, aby uživatele při psaní nezpomalovala.

Kapitola 5

Návrh uživatelského rozhraní

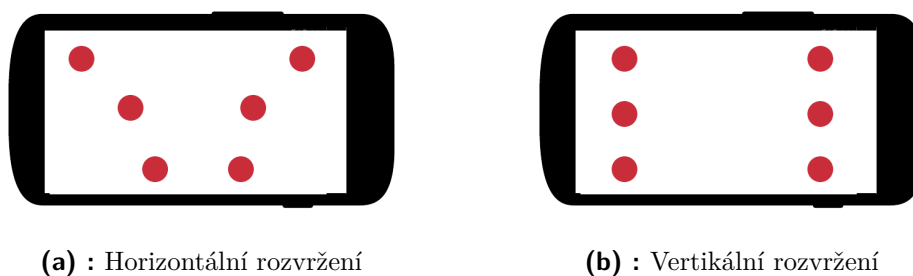
Tato kapitola se zaměřuje na rozhraní klávesnice z pohledu uživatele, konkrétně její vzhled a ovládání. Jedná se o velmi důležitou součást celé klávesnice, protože jak ovládání, tak celkové rozvržení klávesnice bude mít vliv na celkový komfort užívání, o to více v případě nevidomých uživatelů, pro které je klávesnice určena. Jedná se o rozhraní mezi uživatelem a mobilním zařízením, proto je důležité se na tuto část zaměřit stejně, jako třeba na vytvoření samotného modelu psaní a predikcí slov.

5.1 Rozvržení klávesnice

Braillova klávesnice se ve většině případů užívá se šestibodovým rozvržením klávesnice. Proto je při návrhu využíván tento typ klávesnice. Samotné psaní u nevidomých na dotykovém telefonu probíhá ve dvou základních režimech:

- **vodorovná poloha (horizontální rozvržení)** - psaní na podložce
- **svislá poloha (vertikální rozvržení)** - pro psaní v ruce

V každém z těchto režimů je nutné přizpůsobit rozvržení kláves danému způsobu psaní. Mezi oběma rozvrženími se displej automaticky přepíná po



Obrázek 5.1: Rozvržení počátečních bodů kláves

přetočení a překročení hraničního náklonu 5° . V případě vyšší hodnoty se tak přepne rozložení do svislého rozložení kláves. Při úhlu menším než 5° se automaticky přepne režim do vodorovného módu.

Rozvržení pro horizontální a vertikální psaní je odlišné z důvodu odlišného způsobu psaní. Při horizontální pozici, kdy je telefon položen na podložce (stůl, pult, ...), se psaní podobá psaní na klávesnici klasického mechanického Braillova stroje (Pichtův, Perkinsův). Klávesy tak vytvářejí na displeji dva půlměsíce vedoucí od spodního středu. Zůstává i stejné pořadí kláves jako na obrázku??

Oproti tomu psaní při svislém držení telefonu probíhá odlišně. Telefon uživatel drží za boční hrany displejem od sebe. Opět má pod levým ukazováčkem klávesu 1 a pod pravým klávesu 4 jako na mechanickém stroji.

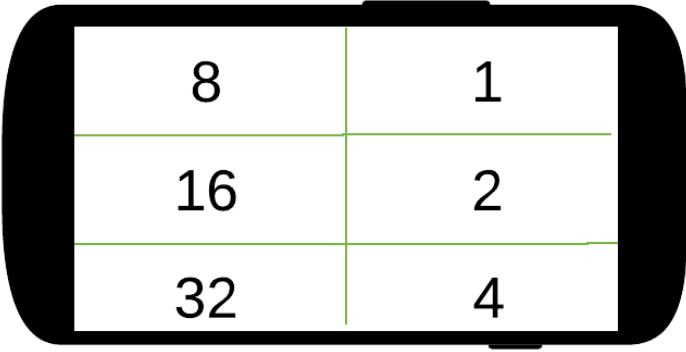
Z úvodního testování a průzkumu mezi nevidomými bylo zjištěno, že pro psaní se využívá téměř výhradně režim displeje v režimu na šířku (landscape). Režim na výšku (portrait) tak bude úplně zablokovan a vždy při zapnutí klávesnice se nastaví pouze tento režim. Je to způsobeno několika faktory. Zaprvé, při držení telefonu na výšku telefon vyklouzává z ruky. Nejde tak bezpečně držet a delší dobu psát, oproti tomu při chycení na šířku telefon sedí pevně v dlaních a lze ho tak snadno ovládat a zároveň mít kontrolu nad pozicí prstů na displeji. Dalším důvodem je i problém při svislém psaní. Uživatel nedrží vždy telefon ve vodorovné pozici a mohlo by tak docházet k přetáčení displeje mezi režimy na šířku a na výšku, když to právě uživatel nechce.

■ 5.1.1 Převod Braillova písma na běžný text

Návrh Braillovy klávesnice tedy obsahuje šestibodové rozvržení, jak je popsáno v části 5.1. Pro převod Braillova znaku na znak klasické abecedy jsem nakonec nepoužil žádnou externí knihovnu (např. Liblouis), ale je využita vlastní implementace překladové tabulky. To znamená pro každý případný

další jazyk vytvořit novou národní tabulku. Při návrhu české převodní tabulky bylo vycházeno z tabulky zobrazené na obrázku 2.5. Ta je spravována organizací SONS.

Při převodu Braillova znaku na symbol je v první řadě nutné získat zakódovanou kombinace stisklých kláves na displeji v podobě jednoho unikátního čísla. Kód (číslo) pak jednoznačně určuje konkrétní znak z abecedy v převodní tabulce. Ta obsahuje dvojici kód-symbol. Kód se ze stisklých kláves vytvoří pomocí součtu hodnot jednotlivých kláves. Každá klávesa má přiřazenou bitovou hodnotu, takže jakoukoliv kombinací kláves vznikne unikátní číslo.



8	1
16	2
32	4

Obrázek 5.2: Hodnota jednotlivých kláves pro sestavení unikátního kódu znaku abecedy

5.2 Ovládání

Tato kapitola se zabývá návrhem toho, jak co nejjednodušeji aplikaci ovládat. Od ovládání samotného psaní až po nastavování aplikace. Jednotlivé prvky je třeba opět přizpůsobit pro nevidomé, nelze pouze využít tradiční postupy.

5.2.1 Psaní textu

Elementární funkce klávesnice. Psaní symbolů probíhá stiskem správné kombinace kláves. Ty jsou ve dvou sloupcích po třech bodech, jak je znázorněno na obrázku ???. Po stisku určité kombinace kláves a následném uvolnění prstů z displeje dojde k hlasové zpětné vazbě. Uživateli je přehráno symbol který vložil do textu, nebo mu je oznámeno, že zadaná kombinace nereprezentuje Braillov znak.

5.2.2 Gesta

Důležitou částí ovládacího rozhraní klávesnice jsou gesta. Ty umožňují uživateli urychlit práci s klávesnicí, ať už při samotném psaní textu, nebo jeho úpravě. Gesta je třeba navrhnout tak, aby jejich provedení mělo logický podklad. To znamená například mezera se vloží tažením prstu ve směru textu a podobně. Zároveň je vhodné zohlednit předpokládanou četnost použití jednotlivých funkcí. To znamená, že pro časté funkce je vhodné volit jednoduchá gesta a naopak pro málo využívané funkce lze využít těžší gesto více prsty. Na druhou stranu gesta pro málo využívané funkce si právě kvůli četnosti použití nemusí uživatel zapamatovat. Proto mají některé funkce alternativní možnost přístupu přes menu, což je rozebráno v další části práce. Návrh typů jednotlivých gest vychází hlavně z úvodního testování, kdy byly nashromážděny požadavky na jednotlivá gesta.

Mezera

Vkládání mezery do psaného textu je jeden z nejčastějších úkonů. Protože Braillovo písmo neobsahuje kombinaci kláves pro mezera, je nutné vkládat mezera jiným způsobem. Na mechanických strojích pro Braillovo písmo, například Pichtově, to je řešeno existencí samostatné klávesy pro mezera. V tomto případě není vhodné přidat speciální tlačítko jako mezerník, protože by to zmenšovalo plochu pro klávesy bodů. Pro mezera je tak využito speciální gesto. Jelikož se předpokládá časté používání, jedná se o gesto tažením pouze jednoho prstu vodorovně přes displej ve směru pomyslného psaní textu, jak je znázorněno na obrázku 5.3. Velikost tahu musí být minimálně 200 px.



Obrázek 5.3: Vložení mezery / potvrzení úpravy

■ Mazání znaku

Dalším úkonem, ke kterému by mělo být přiděleno jednoduché gesto z důvodu častého užívání je mazání znaku. Stejně jako u mezery se jedná o vodorovné tažení prstu v pomyslném směru proti směru psaní textu. Tím dojde ke smazání posledního znaku.

Zároveň je toto gesto využíváno pro odmítnutí autokorekce nebo doplnění slova. Tím se zabýváme v kapitole 7.



Obrázek 5.4: Mazání znaku / zrušení úpravy

■ Přečtení textu

Jednou z nejdůležitějších věcí pro nevidomého uživatele je to, aby mohl mít v jakékoli chvíli přehled o napsaném textu. K tomu slouží opět jednoduché gesto navržené na ovládání jedním prstem, tahem dolů. V takovém případě dojde k přečtení celého napsaného textu od jeho začátku.

■ Posun po slovech

V případě, že chce nevidomý změnit již napsaný text, případně si ho zkontrolovat, má trochu obtížnější orientaci. Zatímco zrakem lze jednoduše procházet již napsaný text, nevidomý se musí neustále textem pohybovat. Proto jsem zařadil gesto na procházení textu po celých slovech, což umožňuje mít přehled o textu, na rozdíl od posunu po písmenech. To se hodí pouze na korekci aktuálního slova.

Tyto gesta, pro posun po slovech, jsou nahraditelná užitím hardwarového



Obrázek 5.5: Čtení celého textu

tlačítka hlasitosti. Stačí se do režimu po slovech přepnout v kontextovém menu. Defaultně je na tlačítko hlasitosti přiřazeno posouvání po písmenech.



Obrázek 5.6: Posun na předchozí slovo



Obrázek 5.7: Posun na další slovo

■ Náповěda slova



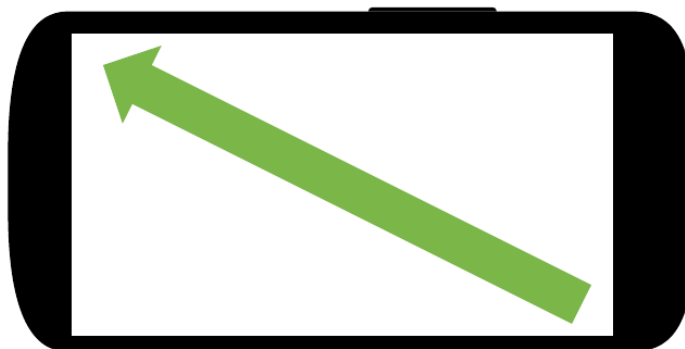
Obrázek 5.8: Doplnění slova pro prefix nebo autocorrect

Gesto pro náповědu slova je tažení prstu ze spodní části displeje do horní části displeje. Akce, které gesto spustí, mohou být dvě. Pokud je slovo rozepsané, to znamená, že poslední zapsaný znak nebyla mezeza, spustí se doplnění nejvhodnějšího slova. To je více popsáno v části 7.2. Pokud žádné slovo není možné doplnit, zůstane slovo rozepsané, ozve se zamítací tón a uživatel pokračuje v psaní.

Druhou možností užití gesta je, pokud se jedná o situaci, kdy bylo právě napsané slovo zvukově označeno jako chybné. To se stane na základě automaticky provedené kontroly autokorekce, která je detailně popsána v části 7.3. V takovém případě se gestem slovo opraví, text-to-speech přečte navrženou úpravu a uživatel může rozhodnout, zda úpravu ponechá, či nikoliv.

■ Smazání textu

Gesto na smazání celého textu znamená, že se veškerý text, napsaný v dané relaci, kdy byla otevřena klávesnice, smaže. Gesto se provede diagonálním tažením prstu z pravého spodního rohu do levého horního rohu. Po smazání může uživatel pokračovat v psaní nového textu, klávesnice zůstává otevřená. Tato možnost zde je pro urychlení mazání textu, aby uživatel nemusel text mazat po jednotlivých slovech.



Obrázek 5.9: Smazání celého textu

■ Kalibrace pozice kláves

Během psaní může dojít k tomu, že se pozice jednotlivých kláves nějakým způsobem změní tak, že se uživateli špatně píše. Například začne vytvářet více chyb a podobně. Někdy se tak může hodit vrácení jednotlivých bodů klávesnice do původní pozice a začít znovu psát. Tato funkce se spustí zatřesením telefonu. Po tom dojde k nastavení obou modelů kláves (horizontální i svislý) k nastavení na jejich defaultní hodnoty.

■ 5.2.3 Kontextové menu

Pokud by všechny úkony, které je třeba provádět pro psaní nebo úpravu textu měly svoje gesto, stalo by se to pro uživatele již nepřehledné, hlavně u funkcí, které jsou málo využívané. Proto existuje z klávesnice přístup do kontextového menu. V něm je možné nalézt jak akce pro které není přiřazené žádné gesto, tak akce které jsou zařazeny v menu jako alternativa pro použití gesta.

Pro vstup do kontextového menu je třeba podržet na displeji v jakémkoliv místě jeden prst alespoň po dobu dvou sekund. Poté dojde k otevření menu, na což je uživatel hlasově upozorněn.

Procházení jednotlivých položek probíhá krátkým stiskem na levé nebo pravé straně displeje pro předcházející nebo následující položku menu. Vždy po přechodu na další položku je přečteno konkrétní znění aktuální položky. Pro potvrzení dané volby slouží opět podržení jednoho prstu na klávesnici po dobu alespoň jedné sekundy. O úroveň zpět se dostane uživatel podržením dvou prstů na displeji, také po dobu jedné sekundy.

Kontextové menu obsahuje tyto položky:

- OK** Potvrzení napsaného textu, jeho uložení a opuštění klávesnice.
- Zrušit text a odejít** Pokud uživatel nechce dále pracovat s textem, chce ho smazat a opustit prostředí klávesnice.
- Po slovech** Přepnutí hardwarového tlačítka na posun textu po slovech. Defaultně je nastaven posun po písmenech.
- Po písmenech** Přepnutí hardwarového tlačítka na posun po písmenech.
- Přečíst text** Přečte veškerý napsaný text během posledního otevření klávesnice.
- Speciální znaky** Otevře klávesnici (menu) pro volbu speciálních znaků.
- Autokorekce zapnuta/vypnuta** Lze přepínat mezi zapnutím a vypnutím autokorekce slov.

■ Speciální znaky

Každá národní verze Braillovy abecedy obsahuje i řadu speciálních znaků (interpunkce, matematické operátory, ...). Bohužel problémem je, že většina uživatelů se s nimi setká pouze výjimečně. Využívané jsou tak většinou pouze základní interpunkční znaménka a některé speciální znaky, jako například zavináč ("@"). Zbylé, které nemají tak časté využití, si tak většina uživatelů nepamatuje a neumí je zapsat pomocí Braillovy abecedy. Proto je kontextové menu doplněno o možnost otevřít si klávesnici speciálních symbolů.

Tato klávesnice funguje na stejném principu jako celé menu. Po otevření může uživatel postupně po jednom znaku prolistovávat skrz celý seznam speciálních znaků, kdy jsou jednotlivé znaky přečteny. Pro potvrzení výběru konkrétního znaku se opět podrží jeden prst na displeji po dobu sekundy a znak se tím vloží na aktuální pozici v textu.

■ 5.2.4 Úprava textu

Text lze v průběhu psaní samozřejmě libovolně měnit i zpětně. Je možné se jakkoliv posouvat v textu oběma směry a měnit při tom napsaný text. Pro pohyb v textu je k dispozici několik možností.

- Gesta
- Hardwarové tlačítko hlasitosti

Pro posun v textu je speciálně vyčleněno hardwarové tlačítko hlasitosti zvuku. Tlačítkem pro zvýšení hlasitosti se posouvá uživatel textem vpřed a snížením hlasitosti naopak vzad. Defaultně je tlačítko hlasitosti nastaveno pro posun po jednotlivých písmenech. Jeho funkce lze ale jednoduše změnit v kontextovém menu. Uživatel si tak může změnit hardwarové tlačítko na posun po slovech. Kromě toho se v textu může pohybovat i jednotlivými gesty.

Kapitola 6

Dynamická pozice kláves

Jak bylo zmíněno v předchozích kapitolách, zásadním problémem nevidomého uživatele při používání dotykového zařízení je absence zpětné vazby o pozici dotyku na displeji zařízení. Musí se tak spoléhat pouze na odhad pozice vzhledem k okraji telefonu nebo na zvukovou a vibrační zpětnou vazbu (ta ale následuje až po napsání znaku).

Při prvotním průzkumu s nevidomými testery bylo zjištěno, že při samotném psaní, právě vlivem absence přesné zpětné vazby o pozici prstu na displeji, se pozice prstů postupně mění oproti prvotnímu stisku. Na to je potřeba při návrhu klávesnice reagovat a průběžně, s pomocí historie stisků uživatele, upravovat pozice jednotlivých kláves na displeji. Cílem tak je, aby se středy kláves přepočítávaly zároveň s tím, jak uživatel v průběhu psaní neúmyslně pohybuje prsty na displeji. Tím se zabrání překlepům způsobených změnou pozice prstů.

Při pohybu ruku po displeji nedochází k tomu, že by se všechny prsty jedné ruky pohybovaly stejně. Například se postupně roztahují, kdy spodní prst míří postupně ke spodnímu okraji displeje, zatímco horní prst k hornímu okraji, nebo otáčejí spolu s rukou. To musí odrážet i model kláves, kdy by se měly jednotlivé klávesy pohybovat nezávisle na ostatních. To je hlavní rozdíl oproti ostatním řešením, které využívají dynamickou polohu kláves. U nich vždy dochází k pohybu celého bloku všech tří kláves jedné ruky. Zároveň je třeba dbát na to, aby nedošlo k situaci, kdy by byla nějaká klávesa vytlačena mimo obrazovku, k čemuž také u již realizovaných řešení dochází. To pak způsobuje situaci, kdy se musí klávesnice znovu zkalibrovat, aby se chybějící klávesy vrátily zpět na obrazovku.

Pro predikci dotyků a úpravě středů kláves je využíváno dvou odlišných přístupů a algoritmů. Jedním je lineární klouzavý průměr, druhý využívá

Kálmánův filtr. Blíže jsou popsány v dalších částech. V obou případech se ale jedná o úpravu pozice jednotlivých bodů odděleně. Vždy, když je konkrétní bod stisknutý, dojde k přepočítání středu konkrétní klávesy.

Individuální model kláves zůstává uložený po celou dobu psaní. Při překlápění displeje během psaní (svislý nebo vodorovný styl psaní) dochází k zachování modelu, jenom se přepne na konkrétní model. Ty se vymažou až při ukončení klávesnice. Při novém otevření klávesnice tak dojde opět k nastavení na výchozí hodnoty jednotlivých středů. Bylo by možné ukládat model při ukončování klávesnice a při novém otevření ho znovu načíst. Problémem ale je, že při každém novém psaní uživatel chytí telefon novým způsobem, takže starý model by neodpovídal úchopu při nové relaci. Navíc se může stát, že by na klávesnici psalo více lidí. Potom by mohlo docházet k nepřesnostem. Proto je vždy nastaven výchozí model.

6.1 Klouzávy průměr

Při využití lineárního klouzávého průměru se bere v úvahu posledních $n = 6$ stisků dané klávesy. Aktuální střed s je tak vždy spočítán podle vztahu

$$s = \frac{1}{n} \sum_{i=0}^{n-1} p_{m-i}$$

kde p_m jsou souřadnice posledního dotyku.

Samotné přiřazení dotyku prstu ke konkrétnímu bodu probíhá na základě velikosti dvoudimenzionální Euklidovské vzdálenosti.

$$d(p, s) = \sqrt{(s_1 - p_1)^2 + (s_2 - p_2)^2}$$

Každému bodu dotyku p je spočítána vzdálenost $d(p, s_n)$ ke všem šesti středům s_a, \dots, s_f . Poté je každému bod přiřazen ke klávese, k jejíž středu má nejbližší vzdálenost. Pokud je pro dva stisky přiřazena stejná klávesa, je to bráno jako jeden stisk.

6.2 Kálmánův filtr

V případě Kálmánova filtru [13] je cílem jeho využití pro predikci polohy uživatelových stisků na základě historie stisků. Opět je vytvořeno šest samostatných nezávislých modelů zvláště pro každý z šesti bodů. Princip s

Kálmánovým filtrem vychází z předpokladu, že pozice uživatelových stisků se mění, ale není úplně náhodná. Uživatel totiž prsty posouvá postupně po displeji s určitou tendencí. Jedná se vlastně o využití Kálmánova filtru na sledování objektu. Objekt je v tomto případě reprezentován uživatelovými dotyky.

Na začátek bylo nutné nastavit inicializační hodnoty pro jednotlivé konstanty Kálmánova filtru. Jejich nastavení bylo určeno experimentálně tak, aby výsledná funkčnost odpovídala požadavkům na chování. Hlavním požadavkem je, aby změna probíhala pomaleji, než v případě klouzavého průměru. Ten je v tomto případě na základě malého počtu dotyků ($n=6$) více dynamický. Také je zapotřebí určit počáteční stavové hodnoty bodu. K tomu je využito defaultní polohy jednotlivých kláves tak, jak bylo určeno jejich rozložení v návrhu pro obě varianty natočení displeje.

Pro každou klávesu lze získat z Kálmánova filtru střední hodnotu a rozptyl z kovarianční matice. Při novém stisku uživatele na displeji tak dojde k tomu, že je pro každou klávesu vytvořena funkce hustoty pravděpodobnosti na základě aktuálního odhadu střední hodnoty a rozptylu. Poté se zjistí relativní pravděpodobnost (pdf) pro každou klávesu a právě stisklé souřadnice dotyku. Bod se přiřadí ke klávese, která má nejvyšší hodnotu pdf.

Po každém stisku dojde opět k aktualizaci jednotlivých filtrů u kláves, ke kterým byl přiřazen stisk. U ostatních zůstává filtr nezměněný. Samotný Kálmánův filtr má dvě základní fáze, predikci a korekci na základě aktuálně naměřené hodnoty. Výstupem filtrace je vektor nezašuměného bodového odhadu x_{n+1} a kovarianční matice P_{n+1} .

Inicializační hodnoty. Na začátku pro spuštění sledovacího procesu je nutné nastavit počáteční stav x_0 a počáteční nejistotu reprezentovanou Gaussovou kovarianční maticí P_0 [12].

$$x_0 = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} a \\ b \\ 0 \\ 0 \end{pmatrix}$$

$$P_0 = \begin{pmatrix} \sigma^2 & 0 & 0 & 0 \\ 0 & \sigma^2 & 0 & 0 \\ 0 & 0 & \sigma^2 & 0 \\ 0 & 0 & 0 & \sigma^2 \end{pmatrix} = \begin{pmatrix} 1000 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 1000 \end{pmatrix}$$

Matice H je matice výběru modelu. Při násobení stavu vybírá pouze prvky, které patří k danému měření.

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Q je odhad chyby procesu. Můžeme předpokládat, že chyba procesu je velmi malá. Tato hodnota zůstává po celou dobu konstantní.

$$Q = \begin{pmatrix} 0.0001 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0.0001 \end{pmatrix}$$

R představuje předpokládanou chybu měření. V případě, že je R vysoké věříme více modelu a měření na predikci nemá takový vliv. V opačném případě, pokud je R nízké, mají měřené hodnoty velký vliv na změnu modelu.

$$R = \begin{pmatrix} 50000 & 0 \\ 0 & 50000 \end{pmatrix}$$

Matice A reprezentuje stavovou přechodovou matici

$$A = \begin{pmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Predikce. Predikce se skládá ze dvou hlavních částí, stavové predikce x_p na základě pohybového modelu a predikce kovariance P_p

$$x_p = Ax_n$$

$$P_p = AP_nA^T + Q$$

Korekce. Poté, co je získáno nové měření (stisk klávesy), může se spustit korekční krok. Ten aktualizuje Kálmánův filtr v podobě aktualizace stavu x_{n+1} a aktualizace nejistoty P_{n+1} .

Inovace \tilde{y} porovnává získané měření s predikcí.

$$\tilde{y} = z_n - Hx_p$$

$$S = HP_pH^T + R$$

Kalman gain K určuje, jak moc věříme predikci.

$$K = P_p H^T S^{-1}$$

Poté už může dojít k samotné úpravě stavu a kovarianční matice.

$$x_{n+1} = x_p + K\tilde{y}$$

$$P_{n+1} = (I - KH)P_p$$

Poté opět proběhne krok predikce. Tyto dva kroky (predikce a korekce) iterativně probíhají k neustálé úpravě stavu na základě sledovaných hodnot.



Kapitola 7

Predikce psaných slov

Jednou z důležitých vlastností klávesnice je predikce slov na základě zadaných znaků nebo celých slov. Je několik možností jak toho využít ve prospěch uživatele a urychlení nebo zpřesnění jeho psaní. Tím, že uživatel zpětně nevidí zadané slovo, může lehce dojít k překlepu, který při psaní nepostřehl. S tím může pomoci autokorekce slov, která by kontrolovala právě zadané slovo a případně by byla navržena úprava daného slova. Tímto tématem se zabývá část 7.3.

Další z možností, jak pomoci s urychlením práce s textem, je vložení možnosti doplnění vhodného slova. To by bylo realizováno na základě zadaného prefixu a uloženého slovníku slov s četnostmi. Může se jednat jak o slovník vytvořený na základě uživatelského vstupu, tedy vytvořený na základě historie uživatele psaní. Postupně by se slovník rozšiřoval a upravoval podle konkrétního uživatele.

Největší problém u tohoto typu úprav spočívá v tom, že uživateli není možné návrhy nebo opravy slov zobrazovat, tak jako se tomu děje na klasické klávesnici. Je tedy nutné se zamyslet nad tím, jakým způsobem uživateli tyto informace sdělovat, aby to naopak nevedlo k jeho rušení při psaní, a tím i zpomalování celého psaní. To by byl přesný opak toho, co je cílem těchto funkcí.

7.1 Český národní korpus

Aby bylo možné vytvořit funkce na autokorekce nebo doplnění slov, musí být k dispozici jazykový základ pro vyhledávání slov. V tomto případě je nutné pro každý jazyk získat seznam slov s jejich četností. Lze to udělat ručně zpracováním velkého množství různorodého textu, nebo jako v tomto případě využít již hotové řešení. V tomto případě frekvenční seznam slov českého jazykového korpusu [23].

Český národní korpus [22] je projekt spravovaný Ústavem Českého národního korpusu na Filozofické fakultě Univerzity Karlovy. Jeho cílem je zpracovávat hlavně elektronický psaný text českého jazyka k jazykovému výzkumu. Český národní korpus tak obsahuje souhrnná data o českém jazyku v podobě rozsáhlých elektronických korpusů. Korpus je soubor autentických textů v elektronické podobě, založený na analýze zdrojů přirozeného jazyka, jako jsou autentické texty (psané i mluvené). S nimi pak lze jednoduše pracovat a vyhledávat různé jazykové jevy související se slovy a slovními spojeními.

Jako základ slovníku pro autokorekci textu i doplnění slov je využit korpus českého psaného jazyka SYN2010 [23] obsahující 100 000 000 slov, rozdělených na 700 000 unikátních tvarů slov. V rámci něj je vytvořen a zpřístupněn frekvenční seznam slov. Ten obsahuje všechny tvary slov z daného korpusu, které mají četnost v korpusu vyšší než 10. Seznam obsahuje dvojice slovo-četnost.

7.2 Dokončování slov pomocí prefixového stromu

Možnost dokončování slov může uživateli urychlit psaní textu. Pokud uživatel ví, že chce psát dlouhé slovo, které navíc již mnohokrát psal, tak si pouze gestem doplní slovo na základě napsaného prefixu (například dva nebo tři znaky daného slova). V ideálním případě se mu konkrétní slovo doplní a pomocí text-to-speech přečte uživateli. Ten se na základě doplněného slova rozhodne, zda je to slovo, které chtěl doplnit. V tom případě napsáním mezery potvrdí slovo a pokračuje v psaní textu dál. V případě, že slovo není to které chtěl doplnit, gestem pro smazání znaku smaže navržený text a zůstane mu jeho počáteční prefix ve kterém může pokračovat.

Při vytváření slovníku používaných slov konkrétního jazyka, v tomto případě českého, je využit Český národní korpus [22], který poskytuje sestavené databáze slov spolu s jejich frekvencí ve srovnávacím frekvenčním seznamu slov. Tím se zabývám v podkapitole 7.1. Slova z této databáze jsem použil pro sestavení datové struktury prefixového stromu, ve kterém se následně vhodná

slova vyhledávají. To, jak probíhá implementace stromu a plnění slovy ze slovníku je rozebráno v podčásti 7.2.1. Jak jsou slova vyhledávána a jaká je efektivita vyhledávání, je znázorněno v podčásti 7.2.2.

7.2.1 Implementace stromu

Srovnávací frekvenční seznam slov českého jazyka je reprezentován pomocí prefixového stromu (trie). Jedná se o acyklický orientovaný graf. V kořeni je obsažen prázdný řetězec. Každý uzel může mít počet potomků až ve velikosti vstupní abecedy.

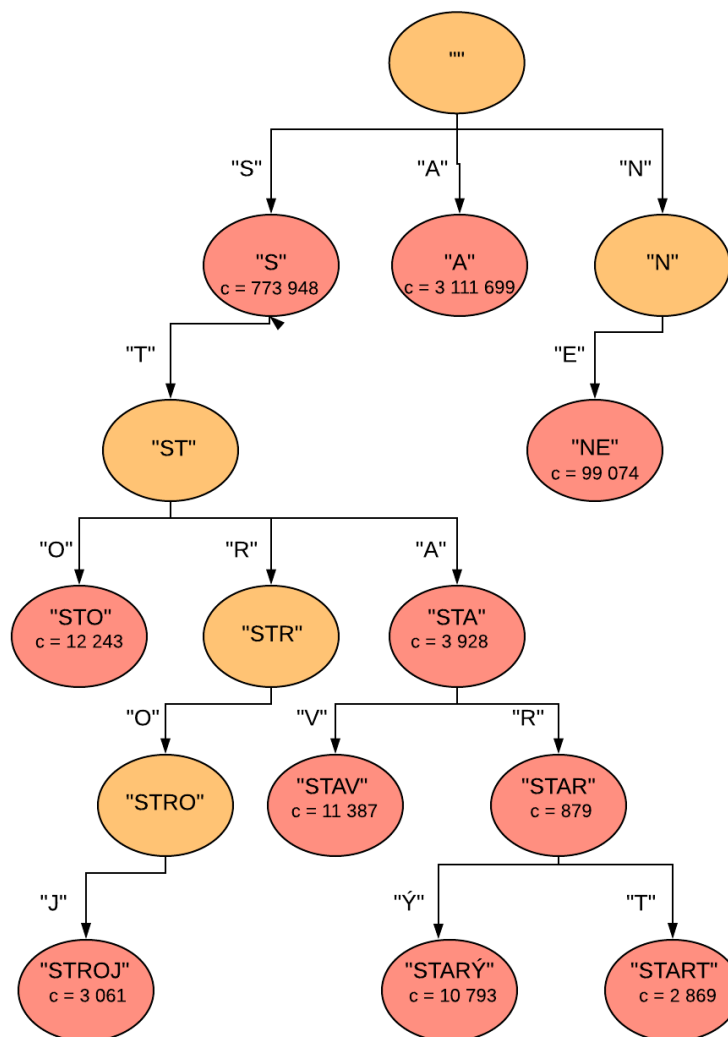
Uzel vždy obsahuje informaci o tom, jestli je daný uzel konečný pro nějaké slovo. Pokud obsahuje kompletní slovo, je v něm informace c o počtu výskytů slova ve frekvenčním seznamu. Při stavbě stromu je také počítán absolutní počet příchozích slov a počet odchozích slov do jednotlivých potomků. To umožňuje ve vyhledávací fázi efektivně prořezávat strom.

7.2.2 Prohledávání stromu

V této části je rozebráno to, jak probíhá samotné vyhledávání korektních slov v již vytvořeném prefixovém stromu. Jak je popsáno v předchozích kapitolách, strom je tvořen uzly, které reprezentují řetězec znaků tvořený ze znaků cesty vedoucí z kořene do konkrétního uzlu. Celý strom obsahuje po nahrání 328 000 slov z frekvenčního srovnávacího seznamu českého jazyka [23] přes 500 000 uzlů.

Samotné prohledávání probíhá od kořene stromu, který neobsahuje žádný znak. V první fázi je třeba projít stromem od kořene přes všechny znaky prefixu k uzlu, jehož cesta od kořene tvoří zadaný prefix. Tím se provede první prořezání stromu. V průběhu této fáze může dojít k tomu, že daný prefix neexistuje. V tom případě se vyhledávání ukončí, vrátí se prázdný řetězec a uživateli se žádné slovo nedoplní, zůstane stále pouze prefix.

V další fázi, poté co byl nalezen uzel, do kterého vede prefix, se hledá samotné nejvhodnější slovo pro daný prefix. Cílem je nalézt takové slovo, které má největší četnost. K vyhledávání využívám prohledávání stromu do šířky. Aby došlo k co největšímu ořezání prohledávaného stromu a nemuseli se procházet všechny uzly, využívám znalosti o zatím nečetnějším nalezeném slovu s daným prefixem.

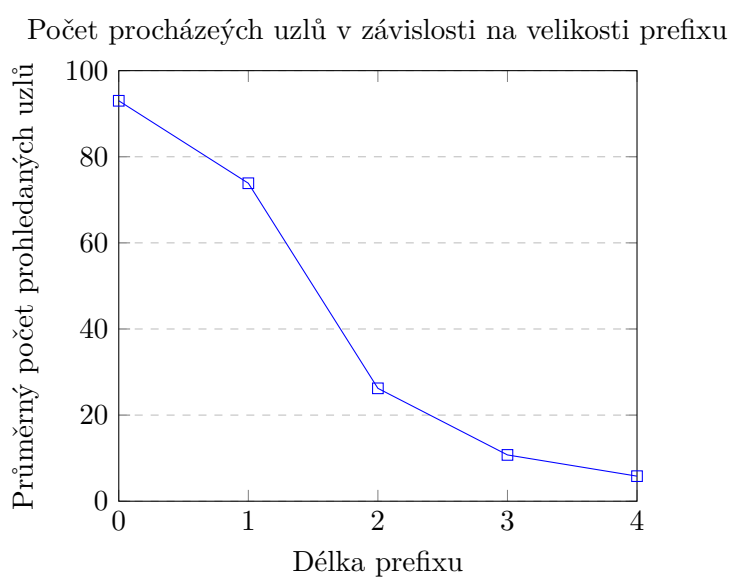
Obrázek 7.1: Prefixový strom s četností slov c

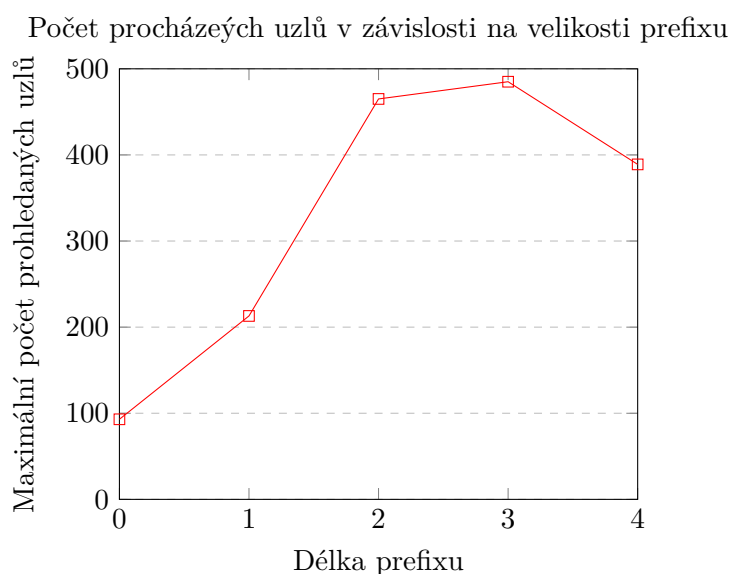
1. Z fronty se odebere uzel. Pokud je uzel cílový uzel pro nějaké slovo, zkontroluje se, zda je jeho četnost vyšší než zatím nejčetnější nalezené slovo. Pokud tomu tak je, změní se odkaz na dosud nečetnější slovo.
2. Do fronty se přidají všichni potomci uzlu X , kteří mají četnost vyšší než dosud nejčetnější nalezené slovo. V dané větvi se může nacházet četnější slovo.
3. Pokud není fronta prázdná, pokračuje se prvním bodem. Pokud je fronta prázdná, prohledávání končí.

Po skončení prohledávacího algoritmu je získán uzel, ve kterém končí slovo se zadaným prefixem a největší četností výskytu.

Vzhledem k vhodnému prořezání stromu, ke kterému dochází ve dvou fázích, nejdříve při vyhledávání uzlu obsahujícího prefix, a poté při samotném prohledávání do šířky, dochází k výraznému omezení v počtu prohledávaných uzlů. V grafu 7.2.2 je zobrazena závislost v počtu prohledaných uzlů a délky prefixu hledaného slova. Pro test jsem vygeneroval všechny kombinace prefixů z české abecedy. Je vidět, že se zvětšující se délkou prefixu klesá průměrný počet prohledávaných uzlů.

Naopak maximálního počtu uzlů se může dosáhnout při délce prefixu 3. Konkrétně se jedná o prefix "neo", pro který to vyhledá slovo "neobyčejně". K nalezení je potřeba projít 485 uzlů, což je stále ale méně než jedno procento velikosti celého stromu. V tabulce 7.1 jsou zobrazeny příklady prefixů různých délek, které vždy projdou největší počet uzlů, než se dostanou ke slovo s největší četností pro daný prefix.





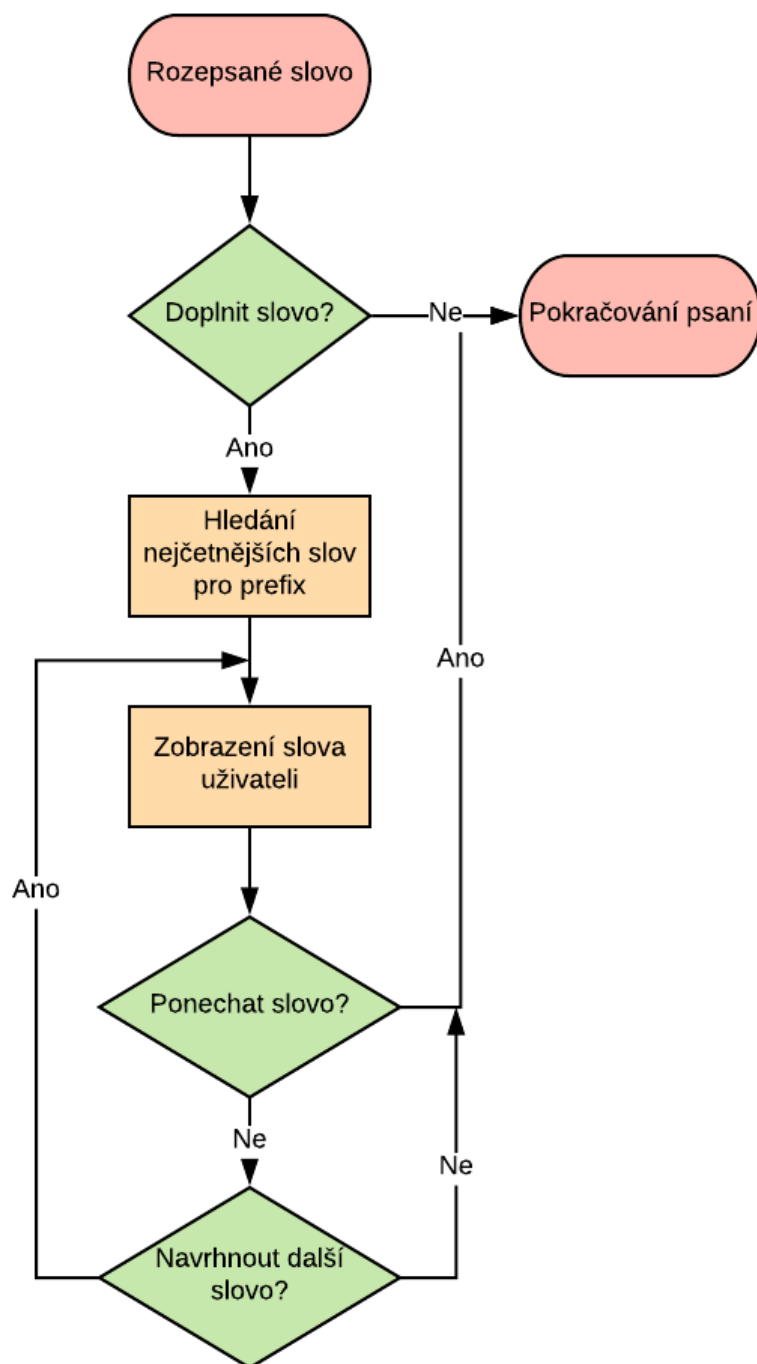
Délka prefixu	Prefix	Nalezené slovo	Počet prohledaných uzlů
0	"	"a"	93
1	"š"	"škole"	213
2	"vy"	"vyšší"	465
3	"neo"	"neobvyčejně"	485
4	"nepo"	"nepochybně"	389

Tabulka 7.1: Prefixy s maximálním počtem procházených uzlů

7.3 Autokorekce slov pomocí BK-stromu

Autokorekce slov přináší uživatelům možnost kontroly slov, zda neobsahují například překlepy nebo vynechaná písmena. Na běžné klávesnici, která obsahuje autokorekci slov je pro uživatele snadno možné reagovat na návrh změny slova. Text je mu zpětně zobrazen a má tak přehled o tom, zda v nějakém slovu došlo k autokorekci či nikoliv. U nevidomých uživatelů je obtížnější poskytnout zpětnou vazbu tak, aby to nebylo obtěžující, ale zároveň měli vždy informaci o tom, že ve slově došlo k možné chybě.

V tomto řešení je tak provedena série kroků, které by měly uživatelům zjednodušit opravu chybného slova, ale zároveň, by měli mít stále přehled o textu. První krok spočívá v kontrole aktuálně napsaného slova. Ve chvíli, kdy uživatel ukončí slovo (vloží mezeru), dojde ke kontrole slova. Kontrola slova probíhá oproti slovům ze synchronního reprezentativního korpusu současné



Obrázek 7.2: Průběh doplnění slova

psané češtiny, tím se zabývám v následující podkapitole. Kontrolou zjistím, zda dané slovo je obsaženo ve slovníku. Pokud je nalezeno, nenásleduje žádná reakce, lze předpokládat, že je slovo správné.

Pokud ovšem slovo nalezeno není, dojde k nalezení nejvhodnějšího náhradního slova. Tím, jak je hledání prováděno, popisuje podkapitola 7.3.1. Zároveň je nutné uživatele upozornit na fakt, že možná došlo v posledním zadaném slovu k chybě. Není možné to provést textově, jako v případě běžné klávesnice. Uživatel je tak upozorněn zazněním chybového zvuku. Na jeho základě se může rozhodnout, zda slovo ponechá. To může nastat pokud ví, že chybný tvar slova zadal záměrně, například pokud se jedná o cizí slovo, zkrácený tvar slova a podobně. Pokud ovšem nic takového úmyslně nenastalo, gestem pro nápovědu slova je uživateli pomocí TextToSpeech přečtena opravená verze slova. Gestem pro mezeru nové slovo potvrdí, gestem pro smazání textu naopak ponechá původní slovo.

Při psaní může být někdy upozorňování na špatné tvary slov pomocí zvuku rušící. Může se jednat například o psaní atypických zpráv obsahující zkratky slov, hovorové tvary a podobně. V takovém případě má možnost uživatel deaktivovat autokorekci slov v kontextovém menu.

7.3.1 Implementace BK-stromu

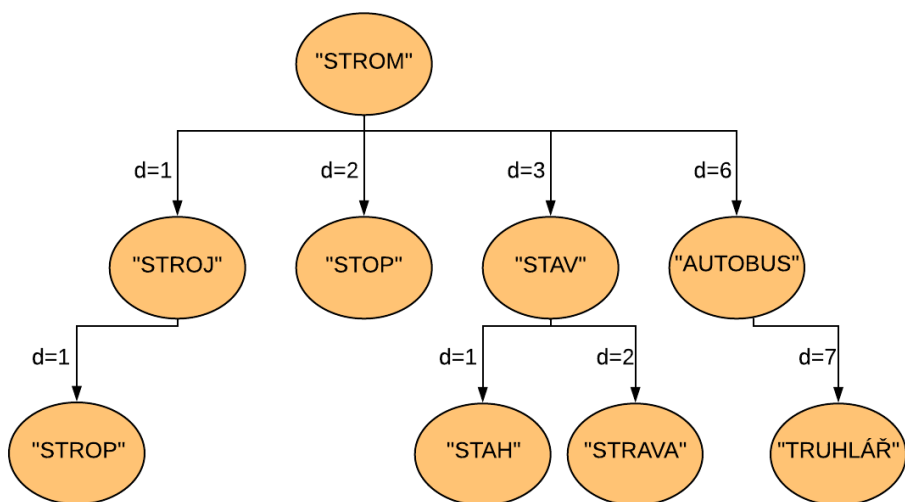
Jako vhodná reprezentace frekvenčního srovnávacího seznamu byl zvolen BK-strom, který představuje vhodnou strukturu pro vyhledávání podobných znakových řetězců [3]. BK-strom, navržený Austinem Burkhardem a Robertem M. Kellerem [6], představuje stromovou strukturu grafu. V každém uzlu je uložen jeden konkrétní tvar slova. V kořeni může být obsaženo jakékoliv slovo. Pro správné sestavení a fungování BK-stromu není nutné kořenové slovo nějak konkrétně volit.

Každý uzel stromu může mít libovolný počet potomků. Rodič je se svým potomkem propojen hranou, která představuje vzdálenost mezi slovem v uzlu rodiče a potomka. Jako metrika vzdálenosti je použita v této implementaci Levenshteinova vzdálenost, ale může být zvolená i odlišná.

Při stavbě stromu je postupováno tak, že se ze vstupního seznamu slov v libovolném pořadí (nemusí být nějak seřazená) vybírají jednotlivá slova. První slovo je kořen stromu. U každého dalšího slova se spočítá daná vzdálenost d (v tomto případě Levenshteinova) mezi vybraným vkládaným slovem a slovem v uzlu. Na základě vypočtené vzdálenosti d se postuje dále. Pokud uzel neobsahuje potomka s hranou vzdálenosti d , potom se vytvoří nový uzel s hranou d , kde nový uzel bude obsahovat vkládané slovo. Pokud ale již hrana s délkou d z uzlu existuje, je nutné rekurzivně postup opakovat. Vkládané slovo se propaguje do potomka do nějž vede hrana s délkou d . V něm se opět spočítá vzdálenost mezi vkládaným slovem a slovem uzlu. Takto se postupuje, dokud se nenarazí na uzel, z něhož potomek se spočítanou délkou d nevede.

Takto sestavený strom tedy obsahuje stejný počet uzlů, jako slov. Každý uzel

kromě vlastního slova obsahuje i další informace. Ty jsou důležité pro následné vyhledávání nejfrekventovanějšího slova nebo ukládání sestaveného stromu pro opakované použití. Konkrétně je v každém uzlu uloženo ID, ID uzlu rodiče, četnost ve srovnávacím frekvenčním seznamu a hodnota vzdálenosti od rodiče.



Obrázek 7.3: Ukázka BK-stromu

Na obrázku 7.3 je znázorněna základní struktura stromu. Při stavbě bylo postupováno v pořadí slov "STROM", "STOP", "STROJ", "STOP", "AUTOBUS", "STAV", "STAH", "STRAVA", "TRUHLÁŘ". V každé větvi tak jsou slova, která mají stejnou vzdálenost od svého předka. To znamená, že například slova "STAV", "STAH" a "STRAVA" mají stejnou vzdálenost od předka "STROM", a to $d = 2$.

Toho se využívá při následném vyhledávání podobných slov ve stromu. Při vyhledávání se určí hodnota tolerance vzdálenosti T , která určuje maximální počet editačních změn mezi výslednými slovy a zadaným slovem. Poté se strom prochází od kořene, spočítá se vzdálenost d_a mezi zadaným slovem a slovem v uzlu. Na list kandidátů podobných slov se přidávají všichni potomci, kteří mají vzdálenost od aktuálního uzlu $d = [d_a - T; d_a + T]$. Pokud se vzdálenost $d_a \leq T$, tak dané slovo v uzlu splňuje podmínku a je jedním z výsledných slov, které jsou v toleranci maximálního počtu editačních změn T .

7.3.2 Průběh autokorekce slov

Samotná autokorekce je vyvolána po každém dokončení slova. To znamená pokaždé, když je v psaném textu vložena mezera, spustí se algoritmus, který zjistí, zda je vložené slovo možné považovat za správné. To se skládá z několika fází:

- zjištění, zda je slovo v existujícím tvaru
- nalezení nejbližšího existujícího slova

Zjištění, zda je slovo v existujícím tvaru

V této fázi je nutné ověřit, zda uživatelem zadané slovo opravdu existuje ve frekvenčním seznamu. K tomu je možné využít více způsobů, které jsou v aplikaci k dispozici. První možností je využití implementovaného prefixového stromu, ve kterém mohou rychle zjistit, zda slovo existuje. Je to stejný postup, jako v první části doplnění slova na základě prefixu. V tomto případě, pokud je možné skrz strom proiterovat od kořene až k poslednímu písmenu slova, znamená to, že dané slovo existuje. Druhou možností je využití samotného BK-stromu. V BK-stromu nalezneme slova, která mají toleranci vzdálenosti $T = 0$. To znamená, že se liší od hledaného slova v maximálně v 0 editačních operacích, takže jsou shodná. Pokud takové slovo s $T = 0$ není nalezeno, strom slovo neobsahuje.

Těmito dvěma způsoby tak lze zjistit, zda zadané slovo je v existujícím tvaru, či nikoliv. Pokud slovo nalezeno není, pokračuje algoritmus dalším krokem, hledáním nejbližšího existujícího slova.

Nalezení nejbližšího existujícího slova

Cílem nalezení nejbližšího existujícího slova je najít takové slovo, které bude mít nejmenší Levenshteinovu vzdálenost od zadaného řetězce. Pokud takových slov existuje více, vybere se takové, které má největší četnost.

Postup je takový, že se rekurzivně postupuje se zvyšující se hodnotou toleranční vzdálenosti T , kdy inicializační hodnota je $T = 1$. Pokud není žádné slovo nalezeno, zvětšuje se T o 1, dokud se existující slovo nenajde.

Uživatel se zvukovým signálem dozví, pokud jeho zadané slovo neexistuje. Pokud zvolí opravu, doplní se mu do textu nalezené slovo, které má možnost přijmout nebo zrušit a ponechat slovo v nenalezeném tvaru.

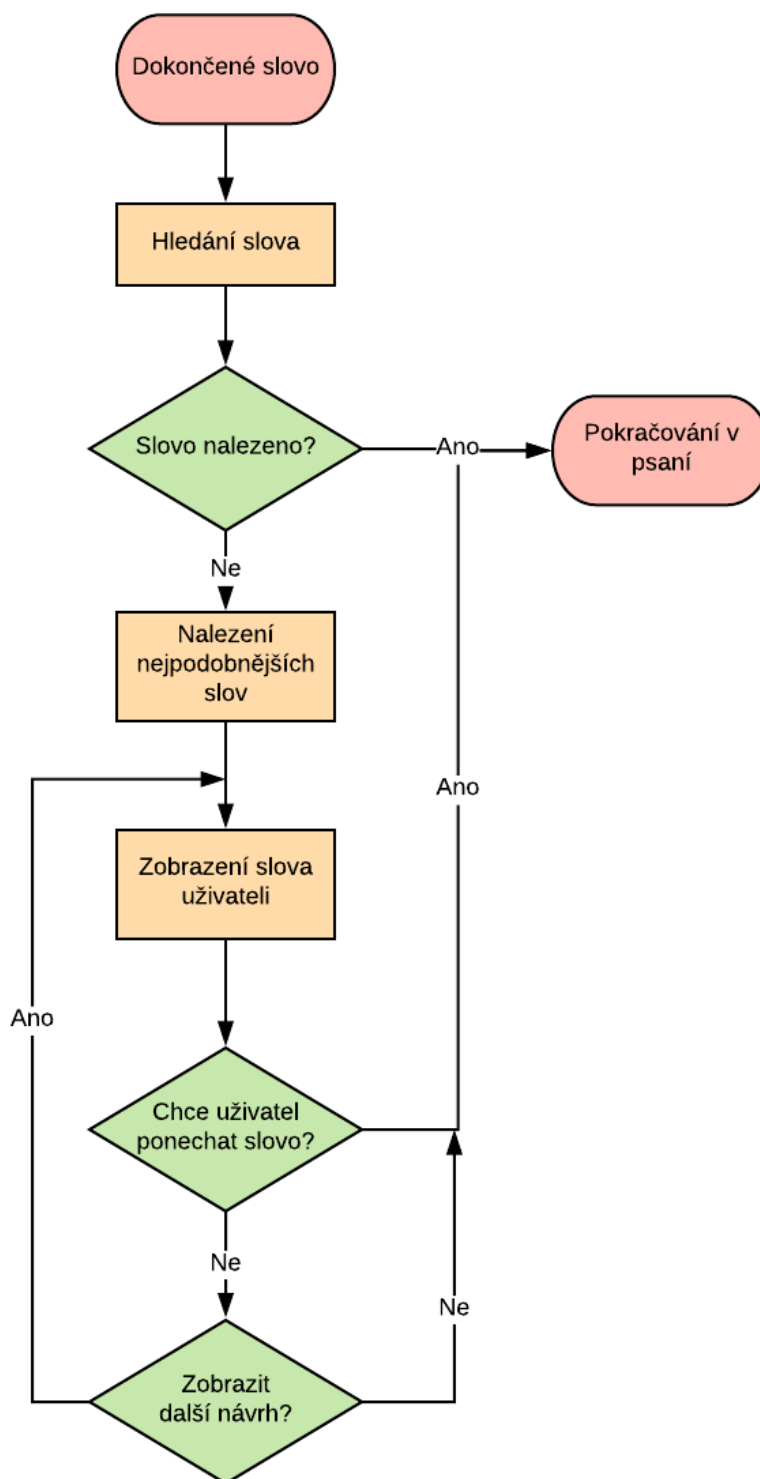
7.4 Ukládání vytvořených stromů

Jedním z problémů při práci s vytvořenými stromy, ať už BK-stromem pro autokorekci nebo prefixovým stromem pro doplňování slov, je jejich uložení pro opětovné použití. Z českého národního korpusu jsem využil frekvenční srovnávací seznamy slov, které jsou k dispozici ve formátu .tsv. Z nich jsem následně sestavil jednotlivé stromy tak, jak jsem popsal v předchozích kapitolách. Ty pak využívám pro efektivní vyhledávání slov. Problém ovšem je, že samotné sestavování stromu ze souboru .tsv je časově náročné, v řádu desítek sekund. To by znamenalo, že by často uživatel dopsal text ještě dříve, než by nějaká korekce slov mohla probíhat. Proto se ze vstupního .tsv souboru vytvoří strom pouze v případě změny souboru (například aktualizace na novější databázi slov). Poté už je soubor ukládán pouze do binárního souboru odkud je i načítám. Tím se celá operace značně urychlí.

Samotné ukládání probíhá tak, že se sestavený strom prochází od kořene vyhledáváním do šířky. Každý nalezený uzel se zapíše do souboru. Cílem je uložen ke každému uzlu pouze takové informace, aby byl výsledný soubor co nejmenší a zároveň obsahoval všechna data potřebná k jeho přesné obnově při načtení ze souboru. Pro každý strom (BK, prefixový) se k uzlu ukládá trochu jiná sada informací:

- prefixový strom (ID uzlu - 32 bitů, ID uzlu rodiče - 32 bitů, počet příchozích slov - 32 bitů, počet odchozích slov - 32 bitů, kód znaku - 16 bitů, ukončení slova - 8 bitů)
- BK-strom (ID uzlu - 32 bitů, ID uzlu rodiče - 32 bitů, vzdálenost od rodiče - 32 bitů, (znak slova - 32 bitů)ⁿ, ukončení uzlu - 8 bitů)

Při následném sestavování se ze souboru čtou data po jednotlivých bytech. Tím, že znám strukturu dat uložených u jednoho uzlu, lze jednoduše znovu sestavit strom do původní podoby v podstatně kratším čase, než při sestavování přímo ze souboru .tsv.



Obrázek 7.4: Průběh opravy slova

Část III

Uživatelské testování

V závěrečné části proběhlo uživatelské testování implementované klávesnice pro Braillovo písmo. Cílem je zjistit, zda realizované řešení odpovídá požadavkům nevidomých uživatelů, jak je uživatelsky přívětivé, a také jaká je efektivita psaní v porovnání s jinými řešeními.

Testování proběhlo ve spolupráci se Sjednocenou organizací nevidomých a slabozrakých ČR (SONS) [17], která pomohla kontaktovat jednotlivé testovací uživatele. Pro testování bylo nutné získat alespoň 3 uživatele, kteří ovládají a mají zkušenosti s psaním pomocí Braillova písma.

Samotné testování se skládá z několika základních částí:

- **Úvodní dotazování** Úvodní fáze má za úkol zjistit o jednotlivých testovacích uživateli obecné informace, díky kterým je poté možné lépe vyhodnotit samotné testování. Dotazování je formou otázek, které se zaměřují kromě obecných informací o uživateli také na jejich zkušenosti v testované oblasti.
- **Testování aplikace** Testování aplikace se skládá z několika samostatných částí. Cílem je, aby se testovací uživatel seznámil se všemi funkcemi aplikace a otestoval je při práci. Součástí je také vyhodnocení účinnosti psaní.
- **Uživatelské hodnocení** Závěrečné uživatelské testování opět probíhá formou otázek, kdy uživatelé hodnotí spokojenost s aplikací, případně navrhuji změny.
- **Vyhodnocení parametrů klávesnice** Zjištění základních parametrů psaní jako chybovost, rychlost psaní.

7.5 Úvodní dotazování

V úvodu testování odpověděli uživatelé na základní obecné otázky týkající se jejich zraku, zkušeností s využíváním mobilního telefonu, Braillovy abecedy. Tyto údaje pak mohou pomoci odhalit některé souvislosti v jejich způsobu psaní.

1. Jaký je Váš věk?
 - a) 0 - 20 let
 - b) 21 - 40 let

- c) 41 - 60 let
- c) 61+ let
- 2. Jaké je Vaše pohlaví?
 - a) muž
 - b) žena
- 3. Jaká je Vaše dominantní ruka?
 - a) pravá
 - b) levá
- 4. Jste nevidomý od narození?
 - a) ano
 - b) ne
- 5. Od kolika let ovládáte Braillovo písmo?
 - a) 0 - 20 let
 - b) 21 - 40 let
 - c) 41 - 60 let
 - c) 61+ let
- 6. Jaký typ mobilního telefonu používáte?
 - a) tlačítkový
 - b) dotykový
 - c) jiný
 - d) nepoužívám mobilní telefon
- 7. Máte zkušenosti s používáním Braillovy klávesnice?
 - a) ano
 - b) ne

7.6 Testování aplikace

Pro testování aplikace je připraveno několik testovacích scénářů. Cílem je, aby uživatel během testování vyzkoušel všechny funkce klávesnice tak, aby je poté mohl ohodnotit. Situace jsou připravené tak, aby co nejvíce odpovídali reálné situaci.

1. Postupně pište text, který je diktován. Pokud si všimnete vzniklé chyby, opravte ji.
2. Seznamte se s funkcemi editace textu. V právě napsaném textu se vraťte zpět a upravte slovo "X" na slovo "Y".
3. Seznamte se s funkcí doplňování slov a vyzkoušejte na slovech.
4. Otestujte funkci automatické opravy slov. Napište libovolné slovo s chybou a následně vyzkoušejte, zda se slovo správně opraví.
5. Otevřete dlouhým stiskem menu, najděte položku speciální znaky a vložte symbol "@".
6. Vyzkoušejte gesto pro přečtení vloženého textu.
7. Gestem smažte celý text.

První bod testování proběhne postupně na více klávesnicích. Metrika jejich psaní se následně porovná a vyhodnotí.

7.7 Zpětná vazba

Tato část zahrnuje hodnocení uživatelů na základě proběhlého testování. Uživatelé hodnotili jednotlivé testované funkce klávesnice kde mohli mít připomínky k jednotlivým funkcím a ohodnotit jejich použitelnost a přínos pro samotné psaní.

Hodnocené kategorie:

- Psaní textu
- Ovládání (gesta i tlačítko)
- Autokorekce slov
- Doplnění slov
- Menu

7.8 Vyhodnocení chybovosti klávesnice

V této části je vyhodnoceno samotné psaní a jeho statistiky. Při uživatelském testování, pokud to bylo možné, byly sledovány tyto parametry psaní

- čas psaní zadaného textu
- počet vzniklých chyb
- typy vzniklých chyb

S pomocí nasbíraných dat bylo provedeno porovnání jednotlivých klávesnic. Metody pro vypočítání přesnosti byly použity z článku [21].

7.8.1 Total Error Rate

První zjišťovanou statistikou je celková míra chybovosti (Total Error Rate) [21]. Ta odráží počet vytvořených chyb INF , i opravených IF , všech typů (substituce, vynechání i vložení znaku) vzhledem k celkovému počtu úderů $C + IF + INF$.

$$TotalErrorRate = \frac{INF + IF}{C + INF + IF} \cdot 100\%$$

Tento výsledek tak určí přesnost během psaní, ale neberu v úvahu to, že si uživatel během psaní některé chyby opraví

7.8.2 MSD Error Rate (minimum string distance)

Další statistikou využitou pro určení přesnosti psaní je MSD Error Rate [20], která je založena na výpočtu Levenshteinovy vzdálenosti MSD [25]. Porovnává se poté Levenshteinova vzdálenost textu předlohy P a výstupního textu T s maximální délkou vstupního nebo výstupního textu .

$$MSD \text{ Error Rate} = \frac{MSD(P, T)}{\max(|P|, |T|)} \cdot 100\%$$

Tato statistika, na rozdíl od Total Error Rate, nebere v úvahu opravené chyby. Pracuje pouze s výsledným textem, který obsahuje pouze neopravené chyby.

■ 7.8.3 KSPC (Key Strokes per Character)

Poslední metrika, která je zjišťována pro porovnání jednotlivých klávesnic je počet provedených úderů (dotyků včetně gest) na vytvoření jednoho znaku výstupního textu. Jde o porovnání počtu správných úderů C , chybných úderů INF , opravených chybných úderů IF , včetně úderů opravující chyby F s počtem správných úderů a neopravených chybných [20].

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

Výsledkem je číslo, které představuje průměrný počet úderů nutných na vytvoření jednoho výstupního znaku (je jedno, jestli je správný nebo špatný). Při tomto výpočtu není brán v úvahu případ, kdy je nějaké písmeno vynecháno.

■ 7.8.4 WPM (Words per Minute)

Statistika sledující rychlost psaní. Představuje rychlost psaní vyjádřenou pomocí průměrného počtu slov napsaných za minutu psaní.

$$WPM = \frac{\text{počet napsaných slov}}{\text{čas psaní v minutách}}$$

Tato statistika se tak vůbec nezabývá tím, jestli jsou při psaní tvořeny nějaké chyby, ale pouze se zaměřuje na čas, za který byl napsán výsledný text.

7.9 Uživatelské testování

V této části je rozebrán průběh testování s nevidomými uživateli, jejich hodnocení aplikace, zpětná vazba a statistika psaní. Uživatelské testování proběhlo se třemi nevidomými uživateli. Podmínkou bylo, aby ovládali a používali Braillovo písmo.

Bohužel se ukázalo, že i přesto, že všichni vybraní uživatelé aktivně používají Braillovo písmo (pro čtení na Braillovém řádku i pro psaní), nejsou všichni schopni používat Braillovu klávesnici na dotykovém telefonu.

V dalších částech jsou zvlášť popsány jednotlivé průběhy testování s každým testovacím uživatelem. Souhrn odpovědí na úvodní dotazování, který dává základní obrázek o uživateli, je součástí přílohy A.1. Všichni vybraní uživatelé mají poměrně dlouho zkušenost s používáním Braillova písma a jsou i uživateli dotykového telefonu. Bohužel se ukázalo, že i přes to jim psaní na Braillově klávesnici nemusí vždy vyhovovat.

7.9.1 Uživatel K

Tomuto uživateli je 63 let a Braillovo písmo využívá od svých 20 let, kdy se stal nevidomým. Aktivně používá u počítače Braillový řádek s osmibodovou klávesnicí. K telefonování využívá speciální dotykový telefon pro nevidomé BlindShell s dotykovou klávesnicí simulující princip tlačítkového telefonu.

S dotykovou Braillovou klávesnicí na mobilním telefonu se dosud nesešel. Byl mu tak vysvětlen princip a možnosti psaní na této klávesnici. Bohužel se ukázalo, že mu koncept klávesnice zcela nevyhovuje. Nemohl telefon pohodlně uchopit tak, aby se cíleně trefoval dotyky na konkrétní klávesy. Během testování zkusil obě možnosti psaní, vodorovně i svisle.

Jedním z problémů, který mu znemožňoval psaní bylo to, že nedokázal zařadit úchop telefonu, takže často při jednotlivých dotycích překračoval plochu displeje na skleněné okraje přední strany (pod kterou již není displej). Bohužel nevidomý uživatel špatně odhalil, že pod sklem již není displej. Proto je důležité najít způsob úchopu tak, aby měl uživatel stále přehled o pozici prstů oproti hranám displeje.

Uživatel dostal na vyzkoušení i jiný typ klávesnice na telefon s OS Android. Konkrétně aplikaci rozebíranou v úvodní části SoftBraille, která má také dynamické klávesy, jak je zobrazeno na obrázku 3.2. To vedlo ke stejnému výsledku jako v případě navržené klávesnice, uživatel nebyl téměř schopný napsat konkrétní znak. Tím se vyloučilo, že by byl problém pouze v této

aplikaci, ale jedná se o obecný problém.

Uživatel také spatřoval problém v tom, že by měl radši jiné rozložení kláves. V případě vodorovného psaní na podložce by mu více vyhovovaly klávesy více vodorovně ke spodní hraně displeje, aby to více odpovídalo mechanické klávesnici. Při svislém psaní by zase preferoval záměnu kláves 1 za 3 a 4 za 6. Což by ale popíralo základní myšlenku toho, že se stejný znak zapisuje kombinací stisků stejných prstů při jakémkoliv způsobu psaní, včetně mechanického stroje. Zřejmě se jedná pouze o nezvyk díky tomu, že s tímto způsobem neměl uživatel zatím žádné dřívější zkušenosti.

7.9.2 Uživatel R

Tato testovací uživatelka je nevidomá od svých 30 let. V té době se také začala učit Braillovo písmo, které aktivně využívá pro psaní i čtení. Telefon používá s dotykovým displejem, ale na něm při psaní používá speciální klávesnici. Braillovu klávesnici na dotykovém displeji nikdy nezkoušela používat.

Na začátku tak byla uživatelka seznámena se základním rozvržením klávesnice a způsobem psaní. Poté si vyzkoušela oba typy psaní, jak na podložce, tak ve svislé poloze. Pro další trénování si zvolila svislou polohu, kdy je telefon umístěn mezi ukazováčky a palci a zbylé prostřední prsty ovládají jednotlivé body na displeji. Po krátkém čase došlo u uživatelky k viditelnému zlepšování. Jediným problémem, na který si stěžovala bylo to, že ji rozložení bodů na displeji přišlo otočené, než jak by si představovala. Je to tak stejný problém, jako popisoval uživatel K.

Vzhledem k průběhu testování a dosažené úrovni psaní nebylo možné přistoupit k testu psaného souvislého textu. Proto byl proveden test na zápis jednotlivých písmen. Jednotlivá písmena byla zvolena tak, aby pokrývala co nejrůznorodější kombinace. Od jednoduchých po obtížné pětibodové. Cílem bylo odhalit, k jakému typu chyb může docházet.

Písmeno	Počet správně	Počet špatně	Popis chyb
a	5	0	-
o	3	2	"m" místo "o", dotyk mimo displej
s	3	2	mimo displej, "!" místo "s"
t	5	0	-
q	5	0	-

Tabulka 7.2: Psaní písmen

Na základě tohoto testu byla identifikována jedna z nejčastějších chyb, ke

Používá ji při psaní na svém iPhone jako primární klávesnici. V úvodu byl při procházení testovacího scénáře postupně seznámen s jednotlivými funkcemi klávesnice.

Na závěr proběhl test psaní souvislého textu (A.3), který byl uživateli diktován. Jeho úkolem bylo pokusit se napsat text co nejpřesněji a nejrychleji. V případě toho, že si všiml chyby si mohl text opravit. Nejprve proběhlo testování s navrženou klávesnicí na mobilním telefonu s OS Android. Poté byl ten samý text napsán pomocí klávesnice iOS Braille Screen Input na telefonu iPhone A.3.1. Z těchto dvou testování byly poté spočítány jednotlivé chybovosti.

Jedním ze základních ukazatelů je Total Error Rate, který zobrazuje poměr chybným znaků, včetně opravených, k celkovému počtu znaků testovaného textu. Zde je vidět, že mírně lepší výsledek je dosažen s navrženou aplikací na Androidu.

$$TotalErrorRate_{android} = \frac{8}{256} \cdot 100 = 3,125\%$$

$$TotalErrorRate_{iPhone} = \frac{17}{256} \cdot 100 = 6,64\%$$

Tato statistika zobrazuje počet úderů nutných na napsání jednoho znaku. Jsou v tom obsaženy i opravy chyb. Opět klávesnice na Androidu dosáhla lepšího výsledku.

$$KSPC_{android} = \frac{276}{256} = 1.078$$

$$KSPC_{iPhone} = \frac{300}{256} = 1.172$$

Tento údaj zase zobrazuje rychlost psaní v podobě počtu slov za minutu. I přes to, že na iOS Braille Input bylo dosaženo větší chybovosti, to znamená, že proběhlo více oprav, rychlost psaní zůstala lehce vyšší než na navržené klávesnici.

$$WPM_{android} = \frac{42}{187} \cdot 60 = 13,47$$

$$WPM_{iPhone} = \frac{42}{183} \cdot 60 = 13,77$$

Jedním z důvodů, proč mohlo být dosaženo na klávesnici iPhone vyšší rychlosti než na klávesnici Android je ten, že iPhone měl nastaveno rychlejší

čtení textu, takže uživatel nemusí tak dlouho čekat na přečtení zpětné vazby (hlavně u oprav, kdy je čtený text delší). Další z důvodů může být ten, že na klávesnici iPhone je uživatel zvyklý psát, přesně zná reakci klávesnice.

Vzhledem k tomu, že si uživatel při psaní vždy všechny napsané chyby opravil, je statistika MSD Error Rate schodná pro obě testované klávesnice.

$$MSDErrorRate_{iphone} = MSDErrorRate_{android} = 0$$

Uživatel také obě verze dynamické změny pozice kláves, jak verzi s Kálmánovým filtrem, tak s klouzavým průměrem. Na výsledné psaní to nemělo pozorovatelný vliv. Je to dáno tím, že většina chyb u tohoto uživatele nevycházela z překlepu mezi sektory, ale například dotyku mimo displej, nebo záměnu písmene.

Na závěr ještě vyzkoušel alternativní klávesnici pro Android, opět SoftBraille. Stejně jako uživatel R si stěžoval na způsob vkládání mezery. Navíc se mi při psaní stalo, že se dostala klávesa 3 mimo obrazovku, takže bylo nutné pozice znovu zkalibrovat. Došlo tak k problému, který byl u této klávesnice popisován v rozboru v části 3.1.2.

7.10 Vyhodnocení testování

Bohužel vzhledem k úrovni psaní jednotlivých testovacích uživatelů nebylo možné vytvořit shrnující statistiku, která by porovnávala jednotlivé uživatele a různé typy klávesnic. Pokud to bylo alespoň trochu možné, provedl jsem dílčí testování. Pro větší detailnější testování a provedení statistik by bylo nutné najít dostatečné množství lidí zkušených s psaním na Braillově dotykové klávesnici, tak jako je například uživatel B, a provést test psaní na dostatečně velkém textu. Najít takové lidi je ale bohužel obtížné.

Předpoklad, že pokud člověk umí psát Braillovým písmem, lehce se přizpůsobí psaní na displeji, úplně neplatí. Jak je vidět ale u uživatelky R, stačilo by chvíli se věnovat tréninku po kterém by bylo možné již klávesnici využívat pro psaní.

Testování uživatele B vyšlo nejlépe, je to díky tomu, že již měl zkušenosti s psaním na dotykovém displeji. I přesto, že aplikace byla na souvislém textu otestována pouze u tohoto uživatele, lze pozorovat to, že psaní není o tolik odlišné oproti klávesnici na telefonu iPhone. Na tomto krátkém textu byla dokonce navržená aplikace méně chybová (Total Error Rate), i bylo potřeba méně úderů na jeden znak (KSPC). A to přesto, že uživatel psal text jako první právě na klávesnici na Androidu. Čas měl naopak mírně rychlejší s klávesnicí na iPhone.

I přesto, že nedošlo u všech uživatelů k vyhodnocení statistik psaní a tím pádem nejde vyhodnotit celková chybovost klávesnice, byla získána během testování celá řada poznatků a důležitá zpětná vazba od samotných uživatelů. V další části jsou rozebrány jednotlivé části aplikace (funkce, rozhraní, ...) z pohledu toho, co odhalili uživatelé, i co jsem vyzoroval u uživatelů.

Psaní textu. Psaní textu funguje dobře. Jak ukázal uživatel B, lze porovnávat navrženou klávesnici rychlostí psaní s klávesnicí na iPhonu, která je zřejmě nejlepší dostupná klávesnice. Podle návrhu uživatele by tam ještě měla být zpětná vazba o právě napsaném slovu pro průběžnou kontrolu textu. Další z věcí, kterou uživatelé během testování navrhli je možnost přizpůsobit si rozložení klávesnice. To znamená pokud teď odpovídá rozložení na stroji, aby si mohli prohodit řádky. Bod 3 místo 1 a 4 místo 6. Tím pak je rozložení jako kdyby se dívali na Braillův znak zepředu a prý by se jim tak lépe psalo. Bylo by tak možné do menu přidat položku, která by měnila mezi jednotlivými rozloženími.

Při psaní byly u jednotlivých uživatelů postupně otestovány oba režimy predikce dotyku. Ukázalo se, že pro začátečníky, kteří ne vždy vědí, kam přesně pokládají prsty se více hodí pomalejší Kálmánův filtr. Pár špatných dotyků tolik nezmění polohu jednotlivých kláves. Pro pokročilejší uživatele, kteří mají zase stisky pod kontrolou je klouzavý průměr dobrý v tom, že relativně brzy reaguje na změny polohy prstů a tím se přizpůsobuje aktuální pozici prstů.

Gesta. Jednotlivá gesta přišla všem uživatelům intuitivní, což byl cíl jejich návrhu. Jediný problém, kterého jsem si všiml byl, že gesto pro mezeru a smazání je nastaveno na velkou dráhu (200 px). Začátečníkům to ani nevadilo, ale uživatel B, který je pokročilý měl někdy problém s tím, že v té rychlosti dělal menší tah, než je minimum a nevložit to tak mezeru. Naopak u začátečníků se někdy objevil problém, že prstem na klávese omylem táhli a tak to vyhodnotilo jako gesto, které přitom nebylo úmyslné. Je to tedy zcela opačná situace v závislosti na úrovni uživatele.

Autokorekce slov. Autokorekce je více využitelná i pro pokročilé uživatele. Tím, že to zvukově oznámí špatné slovo po ukončení chybného slova, všimne si uživatel lehce chyby. Při testování to využíval i uživatel B.

Doplňování slov. To, co jsem si myslel, se potvrdilo. Doplnění slov není vhodné pro pokročilé uživatele, protože ti píšou takovou rychlostí, že se jim

nevyplatí se zdržovat doplňováním slova. Naopak začátečníci a pomalejší uživatelé by tuto funkci mohli rádi využívat, jak potvrdil uživatel K i R.

Menu. Největší problém při psaní byl způsoben otvíráním menu. Netýká se to uživatelů, kteří umějí psát rychle. Je to problém hlavně těch, kteří znaky vytvářejí déle, takže se jim stane, že podrží na displeji jeden prst po stanovenou dobu 2 sekund a tím se otevře menu. Stávalo se to poměrně často u uživatelů R i K, takže zde to určitě chce udělat změnu. Jinak k samotnému fungování menu neměl nikdo připomínky, takže tam není potřeba nic upravovat.

Část IV

Závěr

V práci bylo stanoveno několik cílů. V první části práce jsem se zaměřil na zjišťování obecných principů Braillova písma. Důležitý byl i pohled do historie, protože ukázal, jak vznikly jednotlivé problémy Braillova písma. Například neexistence jednotné tabulky znaků a podobně. Zároveň bylo důležité zjištění toho, jak postupně probíhal vývoj samotného psaní. Od roku 1901, kdy vznikl první mechanický stroj vlastně již nedošlo k žádné výrazné změně zápisu Braillova písma, stále se používá stejné rozložení klávesnice, i když na různých zařízeních.

Kromě vývoje Braillova písma byla stejně důležitá část o současných řešeních, nebo řešeních která byla experimentálně vyvíjena pro dotykové telefony. Na základě toho bylo možné identifikovat problémy, se kterými se tyto práce a řešení potýkali a těm se při vlastním návrhu vyhnout. Je to například špatný design uživatelského prostředí, kdy není využita celá plocha displeje, nebo klávesy mohou opustit obrazovku. Takových problémů byla objevena celá řada.

Na základě informací zjištěných v úvodní části práce tak bylo navrženo řešení šestibodové Braillovy klávesnice. Hlavními vlastnosti jsou využití dynamické polohy jednotlivých kláves, které využívají celou plochu displeje telefonu. Pro dynamickou změnu bylo využito dvou odlišných přístupů, které byly v závěrečné části otestovány. Navrhované řešení bylo ještě před realizací konzultováno s nevidomými uživateli v centru SONS, aby došlo ke zpracování všech požadavků, které by klávesnice měla splňovat. Kromě dynamických kláves to jsou pomocníci pro doplňování slov nebo jejich opravu.

V závěrečné části proběhlo testování realizované aplikace. To ukázalo, že je obtížné provést testování, které by umožnilo vyhodnotit kompletní statistiky jako chybovost psaní, čas psaní a další. Je to způsobeno odlišnou úrovní jednotlivých testovacích uživatelů, ti byli vybráni na základě znalosti Braillova písma, ovšem ne každý se dokázal přizpůsobit psaní na displeji. Ukázalo se ale, že je možné přizpůsobit se relativně rychle psaní na displeji a při pravidelném používání si na ni zvyknout. To bylo ukázáno i u testovacího uživatele, který se již naučil klávesnici používat na iPhone. Po krátkém seznámení s klávesnicí poté navrženou klávesnici používal přibližně se stejnými hodnotami rychlosti i chybovosti. To ukazuje, že klávesnice je srovnatelná s řešením na iPhone. Jde pouze o to, věnovat tréninku nového stylu psaní nějaký čas navíc. Jako užitečné se ukázaly funkce predikce slov (autokorekce i doplňování slov), které pomáhají hlavně začátečníkům, kteří nemají při psaní ještě takovou přesnost.

Závěrečné testování s nevidomými uživateli bylo nakonec přínosné, protože ukázalo problémy pro jednotlivé skupiny uživatelů podle jejich úrovně psaní. Každý uživatel, přinesl na základě své úrovně jiné poznatky k psaní a vylepšení funkcí tak, aby pro danou úroveň uživatele bylo psaní ještě pohodlnější. Pro uživatele, který psal plynule to je například zrychlení čtení textu, než je tam v současnosti, rychlé přečtení právě napsaného slova, nebo zkrácení gesta pro mezeru, aby došlo k dalšímu zrychlení. Oproti tomu v psaní slabší

uživatelé by například ocenili další vylepšení funkcí pro nápovědu slov nebo možnost upravit si rozvržení kláves. Práce tak přinesla nevidomým uživatelům OS Android možnost používat Braillovu klávesnici se kterou mohou zlepšit využívání mobilního telefonu.



Přílohy

Příloha A

Výsledky testování

A.1 Výsledky úvodního dotazování

Otázka	Osoba K	Osoba R	Osoba B
1 věk	c (63)	c (48)	c (53)
2 pohlaví	a (muž)	b (žena)	a (muž)
3 ruka	pravá	pravá	pravá
4 od narození	ne	ne	ano
5 Braille	a (20)	b (30)	a (7)
6 telefon	b (BlindShell)	b (BlindShell)	b (iPhone)
7 klávesnice	ne	ne	ano (iPhone)

Tabulka A.1: Úvodní dotazování

A.2 Výsledky závěrečného dotazování

A.2.1 Uživatel K

Psaní textu. U psaní naležato by bylo lepší, kdyby klávesy byly vedle sebe jako na normální klávesnici, takhle tam nejdou ohnout prsty. U režimu na výšku

Gesta. Gesta takhle jak jsou dávají smysl, takže tam není potřeba nic měnit.

Autokorekce slov. Je to zajímavá funkce, je hned vidět, když je ve slovu chyba.

Doplňování slov. Určitě by to mělo uplatnění. Při psaní bych to občas zkusil použít.

Menu. Při psaní se mi často otevíralo, když jsem nechtěl. Procházení položek je v pořádku.

■ A.2.2 Uživatel R

Psaní textu. Asi bych si na to zvykla, když bych chvíli trénovala. Dovedu si představit, že by se pak lépe psalo v hlučném prostředí, než s klávesnicí kterou používám teď, kde musí člověk poslouchat, na jaký znak zrovna mačká. Jenom mi trochu dělá problém to rozložení kláves, musím se na to hrozně soustředit, ale chce to si asi jenom na to zvyknout.

Gesta. Jak jsou teď udělána, tak jsou intuitivní, takže bych na nich nic neměnila.

Autokorekce slov. Určitě bych to využila.

Doplňování slov. Stejně jako autokorekce se i doplnění může někdy hodit.

Menu. Do menu se možná až lehce dá dostat, párkrát se mi otevřelo při psaní, když to nebylo potřeba.

■ A.2.3 Uživatel B

Psaní textu. Psaní textu se mi líbí, za tu chvíli jsem si na něj zvykl. Je to podobné jako na iPhoneu. Možná by bylo dobré doplnit čtení kompletního slova po jeho napsání.

Gesta. Gesta jsou navržena dobře, jenom by bylo lepší u mezery a mazání zmenšit minimální velikost tahu.

Autokorekce slov. Pro někoho to může být užitečné, ale já bych to nevyužil. Jenom by mě zajímalo, jak by to fungovalo.

Doplňování slov. Doplnění slova by mě naopak při psaní zdržovalo, rychlejší je, když to napíšeš ručně.

Menu. Menu je vymyšleno dobře, jde se do něj snadno dostat.

■ A.3 Testovací text

Jako testovací referenční text byla zvolena úvodní část článku ze zpravodajského serveru iDnes [11].

Testovaný text:

"Na ostrově se po dvanácti letech narodilo dítě. Ostrůvek u Brazílie porody zakázal a všechny tamní ženy posílá rodit na pevninu. Žena, která v sobotu přivedla na svět holčičku, však o svém těhotenství údajně nevěděla a s pomocí přítele porodila v koupelně."

počet slov: 42

počet znaků: 256

počet unikátních slov: 38

■ A.3.1 Text zapsaný na iPhone pomocí iOS Braille Screen Input

Text zapsaný uživatelem B při testovacím psaní pomocí iOS Braille Screen Input. Šipky nad písmeny znázorňují smazání znaku. "Na ostrově se po dvanácti letech narodilo dítě. Ostrůvek u Brazílie porodit zakázal a všechny tamní ženy posílá na pevninu. Žena, která v sobotu přivedla na svět holčičku, však o svém těhotenství údajně nevěděla a s pomocí přítele porodila v koupelně."

■ A.3.2 Text zapsaný na Android pomocí navrženého řešení

Text zapsaný uživatelem B při testovacím psaní podle vzorového textu pomocí navržené klávesnice s využitím klouzavého průměru. Šipky nad písmeny znázorňují smazání znaku. "Na ostrově se po dvanácti letech narodilo dítě. Ostrůvek u Brazílie porodit zakázal a všechny amnní ženy posílá rodit na pevninu. Žena, která v sobotu přivedla na svět holčičku, však o svém těhotenství údajně nevěděla a s pomocí přítele porodila v koupelně."



Příloha B

Obsah CD

Soubor	Popis
DP_Hryzak_Ondrej.pdf	text diplomové práce
DP_Hryzak_Ondrej_android.zip	implementované řešení klávesnice



Příloha C

Literatura

- [1] Liblouis contributors: The liblouis library. <http://liblouis.org/>, květen 2018.
- [2] NV access. Nvda – open source screen reader pro windows. http://nvda-project.cz/?page_id=3, 2018.
- [3] R. Baeza-Yates and G. Navarro. Fast approximate string matching in a dictionary. In *Proceedings. String Processing and Information Retrieval: A South American Symposium (Cat. No.98EX207)*, pages 14–22, Sep 1998.
- [4] Rupert R A Bourne, Seth Flaxman, Tasanee Braithwaite, Maria Vittoria Cicinelli, Aditi Das, Jost B Jonas, Jill Keeffe, John H Kempen, Janet L Leasher, Hans Limburg, Kovin Naidoo, Konrad Pesudovs, Serge Resnikoff, Alex Silvester, Gretchen Anna Stevens, Nina Tahhan, Tien Y. Wong, and Hugh R Taylor. Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis. *The Lancet. Global health*, 5 9:e888–e897, 2017.
- [5] Rupert R A Bourne, Seth R Flaxman, Tasanee Braithwaite, Maria V Cicinelli, Aditi Das, Jost B Jonas, Jill Keeffe, John H Kempen, Janet Leasher, Hans Limburg, Kovin Naidoo, Konrad Pesudovs, Serge Resnikoff, Alex Silvester, Gretchen A Stevens, Nina Tahhan, Tien Y Wong, and Hugh R Taylor. Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis. *The Lancet Global Health*, 5(9):e888 – e897, 2017.

- [20] R. William Soukoreff and I. Scott MacKenzie. Metrics for text entry research: An evaluation of msd and kspc, and a new unified error metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 113–120, New York, NY, USA, 2003. ACM.
- [21] Caleb Southern, James Clawson, Brian Frey, Gregory Abowd, and Mario Romero. An evaluation of brailletouch: Mobile touchscreen text entry for the visually impaired. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '12, pages 317–326, New York, NY, USA, 2012. ACM.
- [22] Ústav Českého národního korpusu FF UK. Český národní korpus: Srovnávací frekvenční seznamy. <http://www.korpus.cz>, 2016.
- [23] Ústav Českého národního korpusu FF UK. Syn2010. <http://ucnk.korpus.cz/syn2010.php>, 2016.
- [24] N. S. Subash, S. Nambiar, and V. Kumar. Braillekey: An alternative braille text input system: Comparative study of an innovative simplified text input system for the visually impaired. In *2012 4th International Conference on Intelligent Human Computer Interaction (IHCI)*, pages 1–4, Dec 2012.
- [25] Soukoreff R. W. and MacKenzie I. S. Measuring errors in text entry tasks: An application of the levenshtein string distance statistic. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems - CHI 2001*, pages 319–320, 2001.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Hryzák** Jméno: **Ondřej** Osobní číslo: **392996**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra kybernetiky**
Studijní program: **Biomedicínské inženýrství a informatika**
Studijní obor: **Biomedicínská informatika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Vývoj vstupní metody pro Android využívající Braillovo písmo

Název diplomové práce anglicky:

Development of an Android Input Method Using Braille

Pokyny pro vypracování:

1. Prostudujte existující literaturu [1] [2] věnovanou implementaci Braillového vstupu na mobilních zařízeních. Seznamte se s tradičními mechanickými metodami psaní v Braillově písmu (Pichtův stroj, Perkins Brailleur) i soudobými softwarovými i hardwarovými řešeními Braillova vstupu a výstupu.
2. Seznamte se s problematikou Braillova písma a jeho implementacemi v různých jazycích. Nastudujte existující možnosti strojového převodu Braille do běžného textu, především nejrozšířenější knihovnu Liblouis [4].
3. Navrhněte a implementujte aplikaci pro Android umožňující zadávat text v Braillově písmu, pomocí kombinací dotyků šesti bodů na displeji telefonu.
4. Otestujte vhodnost navrženého řešení s nejméně třemi nevidomými nebo těžce zrakově postiženými uživateli znalými Braillova písma. Při testování zohledněte existující přístupy popsané v [3].

Seznam doporučené literatury:

- [1] J. Siqueira et al., "Braille Text Entry on Smartphones: A Systematic Review of the Literature," 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Atlanta, GA, 2016, pp. 521-526.
- [2] N. S. Subash, S. Nambiar and V. Kumar, BrailleKey: An alternative Braille text input system: Comparative study of an innovative simplified text input system for the visually impaired/ 2012 4th International Conference on Intelligent Human Computer Interaction (IHCI), Kharagpur, 2012, pp. 1-4.
- [3] Caleb Southern, James Clawson, Brian Frey, Gregory Abowd, and Mario Romero. 2012. An evaluation of BrailleTouch: mobile touchscreen text entry for the visually impaired. In Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services (MobileHCI '12).
- [4] Liblouis contributors: The Liblouis library. <http://liblouis.org/>

Jméno a pracoviště vedoucí(ho) diplomové práce:

doc. Ing. Daniel Novák, Ph.D., Analýza a interpretace biomedicínských dat FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **11.01.2018**

Termín odevzdání diplomové práce: **25.05.2018**

Platnost zadání diplomové práce: **30.09.2019**

doc. Ing. Daniel Novák, Ph.D.
podpis vedoucí(ho) práce

doc. Ing. Tomáš Svoboda, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studenta