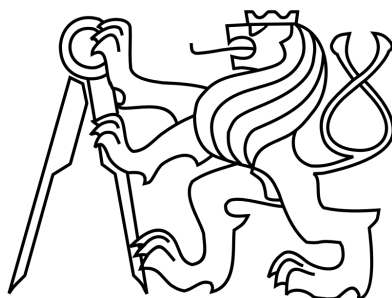


České vysoké učení technické v Praze  
Fakulta elektrotechnická



Bakalářská práce

**Zadávání textu na mobilním zařízení pomocí multitouch gest**

*Daniel Rys*

Vedoucí práce: Ing. Ondřej Poláček

Studijní program: Softwarové technologie a management

Obor: Web a multimédia

Květen 2012



## **Prohlášení**

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

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

V Praze dne 24.5.2012 .....



## **Abstrakt**

Tato práce je zaměřena na průzkum existujících vstupních metod pro mobilní zařízení, následně na vývoj a testování vlastní vstupní metody využívající více dotyků (multitouch) na zařízeních se systémem Android. Práce popisuje vývoj od prototypů až k prvním funkčním verzím, které jsou testovány s uživateli z hlediska intuitivnosti, rychlosti učení metody a podobně. Výsledky testování použitelnosti (usability tests) pak popisují, do jaké míry by taková metoda mohla být použitelná a obstát v konkurenci ostatních, již existujících.

## **Abstract**

This work is dedicated to research of existing text input methods for mobile devices and subsequently on development and testing of an own input method with the usage of multitouch gestures on devices with the Android operating system. The text describes the whole development from prototypes to the first functional versions of the application, which are tested with users to find out their intuitiveness, typing speed, speed of learning the method etc. The results of usability testing describe if the method would be usable in the real life and in competition with other, already existing methods.



## Obsah

1	Úvod .....	12
1.1	Obecný úvod .....	12
1.2	Cíle .....	12
1.3	Organizace dokumentu .....	13
2	Problematika vstupu textu, současné metody .....	14
2.1	Obecné informace .....	14
2.2	Vstup textu na klasických telefonech.....	15
2.2.1	MultiTap .....	15
2.2.2	T9 .....	16
2.2.3	LessTap.....	16
2.2.4	LetterWise.....	17
2.3	Vstup textu na dotykových displejích – stylus.....	17
2.3.1	UniStrokes.....	18
2.3.2	Graffiti .....	18
2.3.3	EdgeWrite .....	19
2.4	Vstup textu na dotykových displejích – kapacitní .....	19
2.4.1	Dotyková QWERTY klávesnice .....	20
2.4.2	Swype .....	20
2.4.3	Vector keyboard.....	21
3	Návrh řešení vstupní metody .....	23
3.1	Cíle .....	23
3.2	Postup návrhu.....	23
3.3	Návrh prototypu.....	24
3.3.1	Expertní vyhodnocení .....	26
3.4	Návrh finální aplikace.....	27
3.4.1	Změny oproti prototypu .....	27
3.4.2	Finální mapování znaků.....	28
3.4.3	Stručný princip – finální aplikace .....	29
4	Implementace .....	31
4.1	Způsob rozpoznání gest.....	31
4.2	Implementace pro OS Android .....	32
5	Dlouhodobá případová studie .....	34
5.1	Popis experimentu.....	34

5.1.1	Způsob testování.....	34
5.2	Cíle testování.....	35
5.3	Setup.....	36
5.3.1	Pomocná aplikace .....	36
5.4	Přehled participantů.....	38
5.4.1	Participant 1.....	38
5.4.2	Participant 2.....	38
5.4.3	Participant 3.....	39
5.4.4	Participant 4.....	39
5.4.5	Participant 5.....	39
5.5	Výsledky testů.....	40
5.5.1	Participant 1.....	40
5.5.2	Participant 2.....	42
5.5.3	Participant 3.....	43
5.5.4	Participant 4.....	44
5.5.5	Participant 5.....	46
5.5.6	Postřehy a připomínky participantů.....	47
5.6	Diskuse.....	47
6	Závěr .....	49
6.1	Budoucí práce .....	49
7	Reference.....	50
8	Příloha - obsah CD.....	52



## Seznam obrázků

Obrázek 1 - Rozložení znaků Multitap .....	15
Obrázek 2 - Rozložení znaků LessTap .....	16
Obrázek 3 - UniStrokes gesta .....	18
Obrázek 4 - Graffiti gesta .....	19
Obrázek 5 - EdgeWrite pomůcka .....	19
Obrázek 6 - Swype .....	21
Obrázek 7 - Vektorová klávesnice .....	22
Obrázek 8 - Prvotní návrh a hi-fi prototyp .....	24
Obrázek 9 - Mapování znaků u prototypu .....	25
Obrázek 10 - Telefon použitý pro testování .....	26
Obrázek 11 - Současný vzhled UI .....	27
Obrázek 12 - Módy klávesnice .....	30
Obrázek 13 - Příklad - různé možnosti pro napsání písmene E .....	32
Obrázek 14 - Zjednodušený diagram tříd .....	33
Obrázek 15 - ukázka pomocné testovací aplikace .....	37

## Seznam tabulek

Tabulka 1 - váhy písmen abecedy .....	28
Tabulka 2 - participant 1, multitouch .....	40
Tabulka 3 - participant 1, GO Keyboard .....	41
Tabulka 4 - participant 2, multitouch .....	42
Tabulka 5 - participant 2, dotyková QWERTY .....	42
Tabulka 6 - participant 3, multitouch .....	43
Tabulka 7 - participant 3, dotyková QWERTY s predikcí .....	43
Tabulka 8 - participant 4, multitouch .....	44
Tabulka 9 - participant 4, dotyková QWERTY .....	45
Tabulka 10 - participant 5, multitouch .....	46
Tabulka 11 - participant 5, dotyková QWERTY s predikcí .....	46

## Seznam grafů

Graf 1 - odhad učících křivek pro participanta 1.....	41
Graf 2 - odhad učících křivek pro participanta 2.....	43
Graf 3 - odhad učících křivek pro participanta 3.....	44
Graf 4 - odhad učících křivek pro participanta 4.....	45
Graf 5 - odhad učících křivek pro participanta 5.....	46



# 1 Úvod

## 1.1 Obecný úvod

V posledních letech zažíváme masový nárůst podílu dotykových telefonů na trhu, a to na úkor klasických telefonů s hardwarovou klávesnicí. Vstup textu na těchto zařízeních je nicméně poněkud problematický, protože dotykové displeje jsou velmi nepřesné z hlediska zadání a zejména na nich chybí dotyková zpětná vazba (uživatel necítí, že stiskl tlačítko), ale zároveň se na ně v případě vstupních metod musí vejít grafické prvky představující celou abecedu a další symboly, ovládání a podobně.

Existuje množství metod, které se tento nedostatek snaží různými způsoby řešit, a to od klasických softwarových klávesnic až po metody využívající kreslení a další pohyby. V této práci se více zaměříme na méně tradiční metody vstupu, které se snaží pro jiný druh zařízení vytvořit i jiné způsoby psaní. Stejnou myšlenkou se bude řídit i výsledná aplikace.

## 1.2 Cíle

Hlavní cíle této práce jsou čtyři – za prvé, vytvořit pokud možno ucelený přehled více i méně tradičních metod vstupu textu se zaměřením zejména na ty, které se týkají zařízení s dotykovým displejem.

Za druhé, vytvořit vlastní metodu vstupu textu, která se od ostatních bude lišit tím, že bude využívat pro zadání více dotyků zároveň nebude vyžadovat takovou přesnost dotyků. Navíc také zjistit, jakým způsobem pracují vstupní metody v systémech Android, v jaké formě můžeme v programu pracovat s dotyky na displeji a podobně.

Za třetí, nově vytvořenou metodu testovat, a to jak v průběhu vývoje (testování prototypů), tak po dokončení prvních funkčních verzí. Testování si klade za cíl, aby výslednou aplikaci mohli vyzkoušet reální uživatelé, kteří nemají žádné hlubší technické vědomosti ani o vstupních metodách obecně, ani o vývoji projektu.

Za čtvrté, z testování dojít k závěru, zda by taková metoda byla použitelná pro širší skupinu lidí a zda poskytuje nějaké výraznější výhody oproti metodám, které již existují a jsou aktivně používány. V případě, že ano, posoudit, zda by stála za další vývoj.

### 1.3 Organizace dokumentu

V první části práce se zaměříme na existující metody, z nichž bude vytvořen celkový přehled. O každé metodě zde najdeme stručný popis a další informace (míra rozšířenosti, typ zařízení na kterém se používá / používala a podobně).

Druhá část bude obsahovat zejména informace týkající se návrhu vlastní aplikace, včetně prototypů, cílové skupiny a předběžné studie použitelnosti. Kromě toho zde budeme moci najít také zdůvodnění a vysvětlení jednotlivých myšlenek a aspektů aplikace (vysvětlení postupu při návrhu).

Ve třetí části najdeme popis finální aplikace (její funkce, vzhled, chování) a stručný popis implementace.

Ve čtvrté, poslední části pak bude popsána dlouhodobá případová studie – popisem experimentu počínaje a výsledky s diskusí konče. V této části bude také popsán průběh testování, jednotliví participanti a další relevantní údaje.

## 2 Problematika vstupu textu, současné metody

### 2.1 Obecné informace

U mobilních telefonů (ať už dotykových nebo klasických) je velký problém v malých rozměrech zařízení a z toho vycházejících malých rozměrech tlačítek a ovládacích prvků. Lidský prst má vzhledem ke své velikosti jen omezenou přesnost, zejména v situacích, kdy má ovládat malá tlačítka. To je problém zejména ve chvíli, kdy se daná osoba navíc nemůže na zadání textu plně soustředit, nebo kdy se při ovládání přístroje pohybuje. Proto se vývojaři jednotlivých vstupních metod snaží, aby tyto nedostatky měly co nejmenší dopad.

Za účelem dělení vstupních textových metod je dobré rozlišit několik hledisek – prvním z nich je to, zda se jedná o vstupní metodu využívající reálná hardwarová tlačítka, nebo dotykový displej. Dalším hlediskem je to, zda se jedná o nějakou formu klasické telefonní 12 znakové klávesnice, o QWERTY klávesnici, nebo zda jde o zcela jiný způsob vstupu textu.

Kategorie způsobů vstupu textu budou řazeny víceméně historicky, od těch nejstarší až po ty, které jsou často používány dnes. V popisech jednotlivých metod, které budou následovat níže, budou vždy obsaženy primárně právě tyto informace, po kterých bude následovat stručný popis účinnosti metody a případně její rozšíření a používání.

Pro posouzení kvality metody se používá několik metrik, v našem případě budou nejdůležitější dvě – a to počet slov za minutu (WPM – Words per minute) a počet stisků na jeden znak (KSPC – Keystrokes per character) [1]. V úvahu budeme brát i náchylnost k chybám.

Pro klávesnice, kde pro každý znak máme jednu klávesu, vychází teoretická hodnota KSPC přesně 1 (jednomu stisku klávesy odpovídá napsání jednoho znaku). Skutečná hodnota se od teoretické vždy liší a jedná se o větší číslo (způsobeno chybnými vstupy).

Výpočet WPM probíhá tak, že spočítáme počet zadaných znaků, vydělíme počtem sekund a celé vynásobíme 60 (aby se jednalo o znaky za minutu, ne za sekundu). Poté vše vynásobíme ještě číslem 1/5. Násobení WPM číslem 1/5 se dělá proto, že se jedná o počet slov za minutu (bez tohoto koeficientu by se jednalo o počet znaků). Hodnota koeficientu je daná normou [1].

## 2.2 Vstup textu na klasických telefonech

Na klasických mobilních telefonech se pro vstup textu používá zpravidla hardwarová klávesnice složená z 12 tlačítek. Mezi nejznámější metody patří zejména MultiTap a T9, zmíníme se ale i o dvou dalších, a to LetterWise a LessTap. Většina těchto metod s klesajícím podílem cílových klasických telefonů ztrácí na významu, nicméně některé z principů zde použitých se používají i dnes.

### 2.2.1 MultiTap

Nejrozšířenější vstupní metoda klasických telefonů s 12 tlačítky. Pro vstup písmen abecedy je využíváno 8 tlačítek, další tlačítka slouží pro zadání speciálních znaků, mezery, mazání znaků a podobně. Rozložení můžeme vidět na obrázku 1. Princip fungování je takový, že každé tlačítko má přiřazených v určitém pořadí (podle abecedy) několik písmen, počet stisků tlačítka v rychlém sledu pak určuje, jaké písmeno bude napsáno.

Tato metoda nedosahuje příliš dobrých parametrů ani z hlediska WPM, ani z hlediska KSPC. WPM odpovídá 7.3, KSPC má hodnotu 2.03 (teoretické), 2.13 (ve skutečnosti) [2].

Kromě těchto nepříliš dobrých hodnot má tento princip zadání další nevýhodu. V případě, že chceme zadat dva znaky, kterým je přiřazena jedna klávesa, musíme po napsání prvního znaku určitou dobu čekat, abychom mohli napsat další znak (doba čekání, neboli timeout, lze odstranit, nicméně pouze za cenu zvýšení KSPC). Timeout je potřebný proto, že pro napsání některých znaků musíme stisknout tlačítko vícekrát, a to právě v rámci této doby.



Obrázek 1 - Rozložení znaků Multitap

### 2.2.2 T9

Svým způsobem kontroverzní vstupní metoda, nicméně taktéž velmi rozšířená. Mnoho uživatelů si na ni zvyklo a dokáží ji efektivně využívat, některým však silně nevyhovuje.

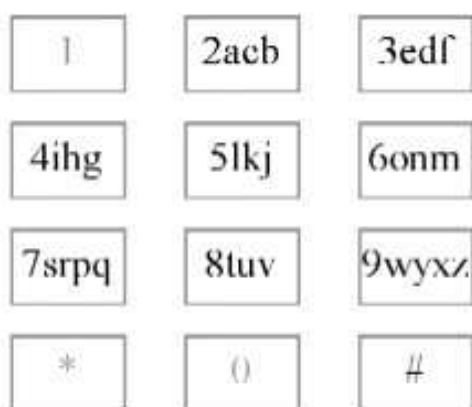
Jedná se o prediktivní metodu, která funguje podobně jako MultiTap s tím rozdílem, že uživatel každé tlačítko stiskne pouze jednou. T9 pak na základě slovníku odhaduje, které ze slov odpovídajících dané sekvenci tlačítek je nejpravděpodobnější. To vede u většiny slov k výraznému zrychlení zadání, nicméně u slov, která nejsou ve slovníku je tato metoda stejně pomalá jako MultiTap.

Parametry T9 jsou ovšem výrazně lepší než u MultiTap, WPM u expertních uživatelů může překročit 20, KSPC (teoretické) odpovídá 1.01 [3].

### 2.2.3 LessTap

Principem zcela totožná s MultiTap klávesnicí, ale jsou zde brány v úvahu i frekvence výskytu písmen abecedy (písmena přiřazená jednotlivým tlačítkům tedy nejsou řazena podle abecedy). To ve výsledku znamená zrychlení zadání. Upravené rozložení můžeme vidět na obrázku 2.

Hodnota WPM zde odpovídá zhruba 7.82, což je oproti MultiTap o necelých 10% více, KSPC ~1.62 (ve skutečnosti)[4]. Jedná se tedy o rychlejší metodu, než MultiTap.



Obrázek 2 - Rozložení znaků LessTap



## 2.2.4 LetterWise

Metoda, která v základní myšlence vychází z metody LessTap, nicméně přiřazení písmen na každém tlačítku není dáno pevně, mění se podle kontextu (pro každou sekvenci znaků je u každého tlačítka vyhodnoceno, jaký znak by mohl následovat s největší pravděpodobností). To výrazně snižuje KSPC a do určité míry i zrychluje zadání zvýšením hodnoty WPM. KSPC této metody pro anglický jazyk odpovídá 1.15, WPM u expertního uživatele se může pohybovat až kolem 20. [13]

## 2.3 Vstup textu na dotykových displejích – stylus

Mobilní telefony, které pro své ovládání používají stylus, nabízejí oproti klasickým telefonům nové možnosti v zadání textu – je ušetřeno místo, které bylo původně určeno pro klávesnici a to může být využito pro větší displej. Stylus jako způsob ovládání telefonu má několik výhod a několik nevýhod.

Mezi hlavní výhody patří fakt, že poskytuje poměrně velkou přesnost (stylus má tenký hrot), takže není problém používat i velmi malé ovládací prvky na displeji (což u dotykových displejů ovládaných přímo prsty je poměrně velký problém).

Mezi hlavní nevýhody lze naopak zařadit hlavně to, že zařízení, kterým k ovládání slouží stylus, nelze ovládat jednou rukou. Kromě toho také to, že stylus je náchylný ke zničení nebo ztrátě.

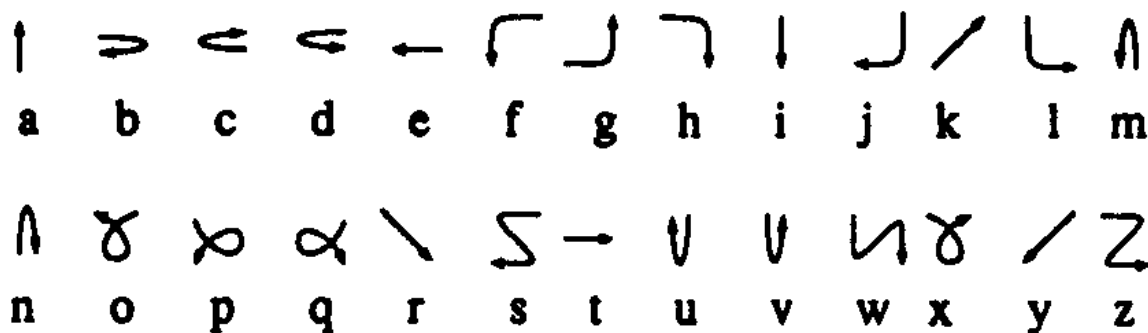
Obecně vzato zařízení ovládaná stylusem silně ustoupila, v mnoha případech přímo vymizela. Některé ze vstupních metod však byly dobře navrženy a obsahovaly zajímavou myšlenku, proto se jim budeme věnovat.

V případě metod, v nichž základní vstupní jednotku tvoří stisk klávesy ale gesto, budeme používat namísto KSPC zkratku GPC (gestures per character), která je obdobou KSPC pro tyto metody.

Pokud pomineme klasickou QWERTY klávesnici, na kterou jsou stylusem vybírána písmena, existuje na tato zařízení hned několik alternativních vstupních metod. Za zmínku stojí zejména UniStrokes, Graffiti a EdgeWrite, o kterých se podrobněji zmíníme níže.

### 2.3.1 UniStrokes

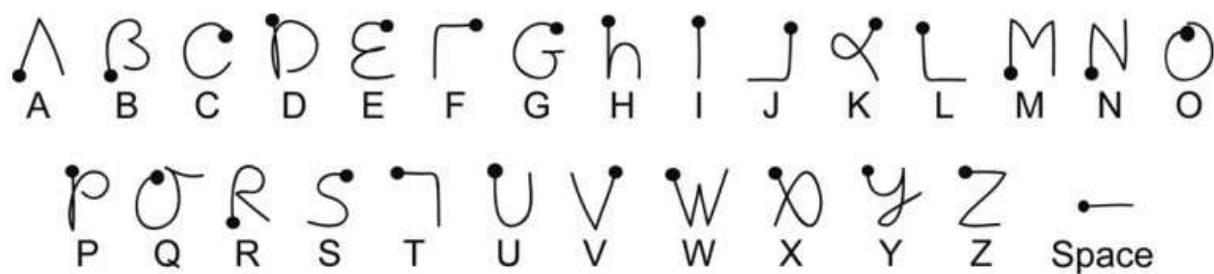
Metoda vstupu textu vzniklá v roce 1993, která používá jednoduchých gest (každé písmeno abecedy má přiřazeno jedno gesto). Nejedná se přímo o kreslení písmen, gesta jsou velmi zjednodušená, nicméně tvůrci této metody se snažili, aby alespoň u některých písmen tvar gesta zhruba odpovídal (toho bylo možné docílit např. u písmen C, O, Z a dalších). Všechna použitá gesta jsou kreslena jedním tahem. Vzhled jednotlivých gest můžeme vidět na obrázku 3. Zároveň zde byla velmi vhodně použita myšlenka, že písmena s větší frekvencí výskytu v jazyce by měla mít jednodušší gesta. Samohlásky a další časté znaky proto dostaly velmi jednoduchá gesta, jako například pohyb jedním směrem. WPM u této metody se pohybuje kolem 15.8 [5]. Teoretické GPC vzhledem k mapování 1:1 odpovídá 1 [6].



Obrázek 3 - UniStrokes gesta

### 2.3.2 Graffiti

Další vstupní metoda založená na kreslení gest na displej, v tomto případě se však objevila snaha o to, aby se gesta co nejvíce podobala písmenům, na něž byla namapována. Z toho vychází vlastnosti, které se od UniStrokes mírně liší – zejména se jedná o snazší učení, na druhou stranu ale pomalejší provádění jednotlivých gest. Kromě toho je základní princip obou metod v zásadě stejný, stejně jako u UniStrokes, i zde jsou všechna gesta prováděna jedním tahem. WPM této metody odpovídá číslu 11.4 [5], mapování znaků na písmena je opět 1:1, GPC tedy opět 1 [6]. Na obrázku 4 je možné vidět jednotlivá gesta používaná v této metodě.



Obrázek 4 - Graffiti gesta

### 2.3.3 EdgeWrite

Princip je zde podobný jako u předchozích metod, opět se objevuje spíše snaha o zjednodušená gesta a možnost větší rychlosti psaní. Přestože jsou gesta zjednodušená, u většiny písmen tvar gesta alespoň vzdáleně připomíná tvar písmene, na které je mapováno. Rozdílný je u této metody princip, jak jsou gesta rozpoznávána. To se neděje pomocí rozeznávání tvaru gesta jako u předchozích dvou metod, ale podle toho, v jakých rozích zadávací plochy a v jakém pořadí se stylus vyskytne (pro zadávání se používá pomocné okénko, které je vidět na obrázku 5). To ve výsledku pomáhá přesněji rozpoznat, co chtěl uživatel zadat a předcházet chybám. Podle některých zdrojů [7] může s touto metodou expertní uživatel psát rychlostí přes 20 WPM při minimálním množství chyb a oprav (a tedy s GPC blízcím se 1) [8].



Obrázek 5 - EdgeWrite pomůcka

## 2.4 Vstup textu na dotykových displejích – kapacitní

Společně s klasickými telefony v současnosti nejrozšířenější segment trhu. Zatím vše nasvědčuje tomu, že podíl těchto zařízení do budoucna ještě poroste. Z hlediska ovládání

poskytuje tento typ telefonu poměrně mnoho výhod (extrémní jednoduchost a intuitivnost), nicméně z hlediska vstupu textu zde narazíme spíše na nevýhody. Mezi tyto nevýhody patří obecně zejména velmi nízká míra přesnosti a fakt, že prst jako ovládací prvek má vzhledem k velikosti displeje také nezanedbatelnou velikost. Odpadá tím možnost použití malých tlačítek a podobných prvků.

Přes tyto nevýhody se tato zařízení velmi rozšiřují, jak bylo řečeno výše, proto se zde budeme věnovat již existujícím vstupním metodám pro tyto mobilní telefony a pokusíme se zjistit, které jejich vlastnosti by byly žádoucí pro lepší způsob vstupu textu.

Níže se v přehledu budeme věnovat metodám jako softwarová QWERTY klávesnice (z hlediska myšlenky se nejedná o nic zásadního, nicméně má vysokou míru používání a uvedeme ji zejména pro srovnání), Swype, Vector Keyboard (vektorová klávesnice).

#### **2.4.1 Dotyková QWERTY klávesnice**

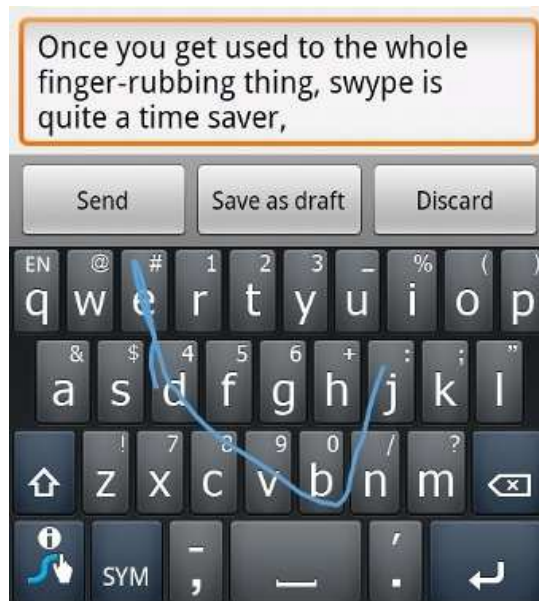
Velmi často používaná metoda vstupu textu, přestože ve své základní variantě není rozhodně ideální. Vyžaduje totiž pro jednotlivá písmena velké množství tlačítek, což ve výsledku v kombinaci s velikostí displeje obvykle znamená, že tato tlačítka budou velmi malá. Další nevýhodou je nedostatek zpětné vazby – obvykle mají tyto klávesnice nějakou formu akustické zpětné vazby (zvuk) a někdy také vibrační, nicméně u tohoto principu psaní se jako lepší z hlediska pohodlnosti i použitelnosti jeví klasická tlačítka klávesnice. Rychlost psaní na dotykových QWERTY klávesnicích může u expertních uživatelů pohybovat až kolem 60 WPM [9], u většiny uživatelů to je však výrazně méně. Rychlost může v některých případech zvyšovat i to, že QWERTY klávesnice pro dotykovou obrazovku existují v různých modifikacích (například s predikcí slov), což uživateli usnadňuje práci.

#### **2.4.2 Swype**

Swype představuje klávesnici s tlačítky (obvykle klasické QWERTY rozložení, existují však i jiná), kde většina psaní neprobíhá klasicky stisky jednotlivých tlačítek, ale tahy přes celou klávesnici. Rozpoznání toho, jaké slovo chtěl uživatel napsat, probíhá tak, že se vyhodnotí, na kterých písmenech změnil směr pohybu (to se považuje za písmeno, které chtěl uživatel napsat) a v kombinaci se slovníkem se odhadne nejpravděpodobnější slovo. Za předpokladu, že je k dispozici kvalitní slovník, je tato metoda velmi efektivní, rychlá, a co je důležitější,

nevyžaduje tak vysokou míru přesnosti, jako například dotyková QWERTY klávesnice. Jediná nevýhoda je ta, že pokud potřebujeme napsat slovo, které se nenachází ve slovníku, jsme odkázáni na psaní klasickým způsobem (jako na QWERTY klávesnici).

Swype pro valnou většinu uživatelů představuje snadnou možnost rychlejšího zadání textu na dotykových displejích – průměrný uživatel Swype dokáže psát stejnou rychlostí (až kolem 40 WPM), jako expertní uživatel dotykové QWERTY klávesnice (jak je uvedeno výše, tam se maximální hodnoty pohybují okolo 60 WPM, to se ovšem týká minima uživatelů)[10]. Na obrázku 6 je vidět layout klávesnice používané u metody Swype.



Obrázek 6 - Swype

### 2.4.3 Vector keyboard

Klávesnice vyvinutá na Elektrotechnické fakultě FEL ČVUT [11]. Od předchozích metod se výrazně liší, protože nepoužívá žádná tlačítka a podobné prvky, ale obsahuje pouze zadávací plochu. Zadání jednotlivých znaků pak probíhá pomocí určování směru pohybu zadávacího bodu (prst, stylus). Tyto směry jsou vyhodnocovány jako vektory, odtud vektorová klávesnice. Vstup každého znaku probíhá tak, že uživatel táhne prstem (nebo stylusem) po určité části zadávací plochy v určitém směru. Jednotlivé kombinace ploch a směrů pak jednoznačně určují výsledný znak (jde o mapování 1:1). Na obrázku 7 můžeme vidět vzhled vektorové klávesnice společně s mapováním znaků.



Obrázek 7 - Vektorová klávesnice

Vstupní metoda navrhovaná a testovaná v této práci má mnoho společných rysů právě s touto vektorovou klávesnicí (základní princip funkce je téměř totožný [11], s tím rozdílem, že zde se bude jednat o verzi schopnou rozlišit více dotyků a pracovat s nimi za účelem zvětšení počtu možných znaků tak, abychom nemuseli zapojovat více směrů nebo více zadávacích ploch).

### 3 Návrh řešení vstupní metody

Při návrhu hrálo velkou roli několik zadaných i jiných podmínek – cílová skupina pro výslednou aplikaci, platforma a zejména snaha o to, aby byla vytvořena metoda, jejíž použití nevyžaduje přílišnou přesnost od uživatele (účelem je vyhnout se tomu, aby se uživatel musel soustředit na to, aby vůbec prstem trefil ovládací element nebo v našem případě správný směr). Jak již bylo řečeno výše, návrh v některých rysech vychází z Vektorové klávesnice [11].

#### 3.1 Cíle

Vytvořit aplikaci, která by plně vyhověla cílové skupině a umožnila komfortnější psaní na dotykových displejích, které by nevyžadovalo zvláštní soustředění ani přesnost dotyků. Z toho vychází i další ze základních cílů, a tím je co nejvyšší míra jednoduchosti. Od začátku zde tedy byla snaha o to, aby zadání nešlo cestou kreslení složitých gest a podobně. Jako nejlepší řešení se ukázala možnost použít pouze jednoduché směry, která dostala konkrétnější tvar po prostudování některých předchozích prací na fakultě [11][12].

Jako cílová skupina, pro kterou je klávesnice primárně určena, byli zvoleni mladí, netechnicky vzdělaní lidé. To nám umožňuje při navrhování vstupní metody předpokládat, že uživatelé nebudou mít problém rychle se učit novým věcem. A to i přes fakt, že výsledná aplikace nebude podobná v podstatě žádné obecněji rozšířené vstupní metodě.

#### 3.2 Postup návrhu

Primární fáze návrhu aplikace probíhaly iterativně – po malých částech, a vždy bylo bráno v úvahu to, zda je vše v souladu s primárními cíli. Nejprve byl zformulován základní princip funkce, poté byly vytvořeny první low-fidelity prototypy a schematické návrhy toho, jak by aplikace mohla po dokončení vypadat. Na těchto schématech a prototypy byl postupně vybudován high-fidelity prototyp, od jehož podoby se odvíjí i finální aplikace.

### 3.3 Návrh prototypu

Po nastudování základních principů týkajících se návrhu uživatelských rozhraní a práce s nimi se podařilo vytvořit prvotní návrh principu funkce a papírový prototyp. Součástí tohoto návrhu byla i sada základních gest, které bylo možno mapovat na znaky.

Základní princip je podobně jako u vektorové klávesnice v tom, že uživatel tahá prsty po zadávací ploše, z čehož vznikají gesta, která můžeme mapovat na znaky. Již v prototypu je zadávací plocha rozdělena na dvě části (levou a pravou) oddělené svislou čarou, přičemž v každé z částí rozpoznáváme čtyři směry. Tažení jedním prstem daným směrem znamená v jednotlivých částech různé znaky. Vzhledem k tomu, že toto by stačilo pouze na 8 znaků, přidáváme možnost použití dvou prstů zároveň (multitouch). Tím se nám při použití vždy jen jedné poloviny a zachování původních směrů zvyšuje počet znaků na 16. To však stále nestačí, proto se dostáváme k tomu, že bude možno použít obě části současně, a co je důležitější, také to, že oba prsty se nemusí pohybovat stejným směrem. To nám umožnilo získat dostatek kombinací na pokrytí celé abecedy a několika dalších metaznaků (mezera, mazání a podobně). Tento základní princip společně s většinou gest základní sady se přesunuly i do finální verze aplikace, kde nicméně nastalo mnoho jiných změn.

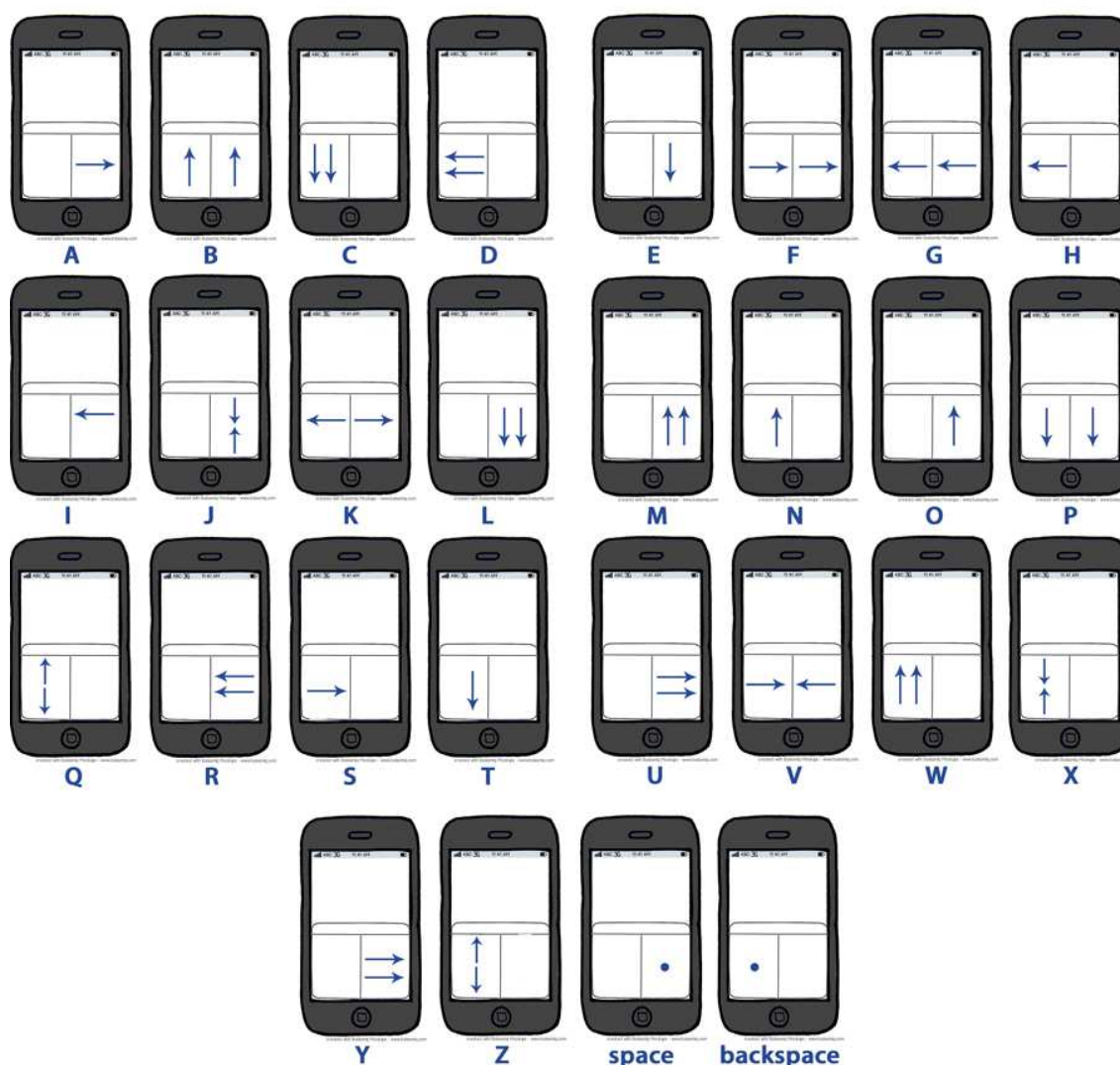
Součástí low-fidelity prototypu byly i tlačítka v rozích pro různé další funkce jako emotikony, shift a podobně, to však bylo ve výsledné aplikaci nakonec vyřešeno jinak. Stejně tak horní lišta, která měla podle prototypu sloužit k přepínání jednotlivých druhů klávesnice (písmena, znaky, čísla) ve výsledné aplikaci není.



Obrázek 8 - Prvotní návrh a hi-fi prototyp



Na obrázku 8 můžeme vidět vlevo prvotní „papírový“ prototyp (neboli spíše nástin toho, jak by měl výsledek vypadat), vpravo pak první funkční verzi, Hi-Fi prototyp, se kterým probíhala první vyhodnocení. Mapování jednotlivých gest na konkrétní znaky se několikrát v průběhu vývoje měnilo, finální verze bude popsána v části věnované implementační fázi návrhu. V prvotním prototypu je obsažena většina gest použitých i ve finální verzi, nicméně některá byla nahrazena (většinou z důvodu, že při testování se ukázala nepohodlnost nebo špatná použitelnost těchto gest). Samotné přiřazení jednotlivých znaků bylo u prototypu zprvu alfabetské (nejjednodušší gesta pro začátek abecedy, obtížnější pro konec) a postupně se upravovalo. Obrázek 9 představuje rozpis mapování znaků v prvotní verzi.



Obrázek 9 - Mapování znaků u prototypu

### 3.3.1 Expertní vyhodnocení

S použitím Hi-Fi prototypu bylo vykonáno expertní vyhodnocení mapování gest na písmena, jehož cílem bylo zjistit, zda se vývoj ubírá správným směrem. Dále mělo za úkol zjistit, od kterých prvků případně v dalším vývoji upustit, a kterých se naopak držet.

Tato studie byla vykonána s dvěma reálnými uživateli, kteří měli po seznámení se vstupní metodou za úkol posoudit, do jaké míry je princip zadání pochopitelný, intuitivní a do jaké míry vzhled uživatelského rozhraní prototypu pomáhá uživateli v tom, aby pro požadovaný znak zvolil správné gesto. Oba tito uživatelé byli znalí problematiky uživatelských rozhraní a náležitostí testování použitelnosti, proto výsledek této předběžné studie pomohl výrazně usnadnit odhalení a napravení některých nedostatků aplikace.

Toto předběžné testování probíhalo na přístroji LG Optimus Black (P970) s operačním systémem Android verze 2.2. Telefon můžeme vidět na obrázku 10. Po celou dobu testování měli uživatelé k dispozici papírový rozpis gest a znaků (grafické znázornění na prototypu ještě nebylo dořešeno – viz obrázek 8). Pro studii byl použit první funkční prototyp vstupní metody, uživatelé měli jednoduchý úkol – napsat celou abecedu a několik dalších vět.



Obrázek 10 - Telefon použitý pro testování

Ze studie vzešlo několik závěrů a doporučení. Nejdůležitějším byl požadavek, aby výsledná aplikace byla výrazně intuitivnější a aby uživatelské rozhraní dávalo dostatečně najevo, které

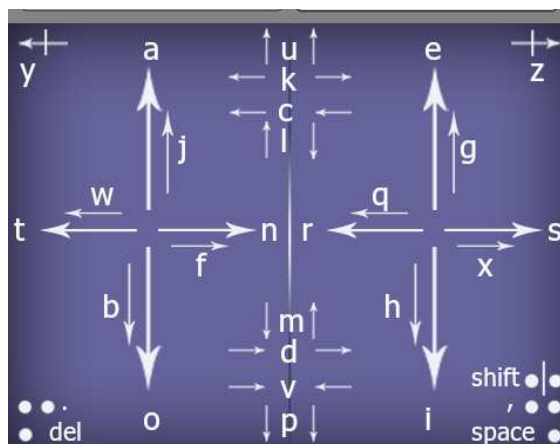
gesto je přiřazeno kterému znaku. Kromě toho se také zjistilo, že některá gesta jsou velmi obtížně nebo nepohodlně vykonatelná a měla by být nahrazena jinými (toto se týkalo u obou uživatelů stejných gest). Poslední problém, který se touto studií podařilo odhalit, byl ten, že některá gesta jsou snadno vykonatelná, pokud uživatel používá dva palce, zatímco jiná lze vykonat, pokud uživatel používá dva prsty jedné ruky (doporučení proto znělo, aby byla vybrána jen jedna z variant). Z toho vyšel také závěr, že všechny tyto nedostatky je třeba upravit a zlepšit.

### 3.4 Návrh finální aplikace

Vzhledem k tomu, že základní funkčnost byla již hotova v prototypu, pro finální aplikaci se mohla veškerá práce soustředit na zdokonalování rozpoznání gest a zejména na záležitosti kolem zpětné vazby a uživatelského rozhraní.

#### 3.4.1 Změny oproti prototypu

Prvotní funkční verze oproti prototypu doznala množství změn. Některé z nich byly vykonány na základě předběžné studie použitelnosti, jiné byly vykonány z důvodu zjednodušení aplikace a zvýšení míry intuitivnosti. Je kladen větší důraz na to, aby se co možná největší množství operací se vstupní metodou dělo právě gesty. Na obrázku 11 můžeme vidět, jak vypadá aktuální UI aplikace (po zapracování změn zmíněných níže).



Obrázek 11 - Současný vzhled UI

Zde následuje výčet nejdůležitějších změn:

- > Bylo upuštěno od záměru implementovat vrchní lištu (přepínání klávesnice vyřešeno speciálním gestem přes dělicí čáru)
- > Stejně tak byla i tlačítka v rozích nahrazena gesty
- > Pro nejčastější operace byla přidána nová gesta – dotyk, dotyk dvěma prsty, dotyk na obou polovinách zadávací plochy
- > Styl zadávání gest byl určen na zadání dvěma palci a gesta tomu byla přizpůsobena
- > Upravena gesta pro některá písmena tak, aby byla snadněji vykonatelná
- > Změněno mapování znaků
- > Vytvořeno grafické UI, jehož účelem je, aby uživatel viděl, jak napsat jednotlivé znaky
- > Popisky na zadávací ploše změněny na dynamické (mění se v závislosti na caps locku a také na tom, jaká klávesnice je právě aktivní – číselná, písmenná...)

### 3.4.2 Finální mapování znaků

Vzhledem k nelogickému rozložení znaků u prototypu bylo při výsledné implementaci mapování výrazně změněno. Pro celou abecedu byly pomocí frekvenční analýzy jazyka [14] určeny váhy jednotlivých písmen (příčemž váhy písmen s diakritikou byly přičteny ke stejným písmenům v základní formě). Váha písmene CH, které na klávesnici samozřejmě jako samostatné písmeno neexistuje, byla rozdělena mezi C a H ve stejném poměru, v jakém se vyskytují v českém jazyce. Rozpis vah písmen můžeme vidět v tabulce 1.

Tabulka 1 - váhy písmen abecedy

A	8,4	N	6,6
B	1,5	O	8,7
C	3,2	P	3,4
D	3,6	Q	0,0
E	10,6	R	4,9
F	0,3	S	5,3
G	0,3	T	5,8
H	1,8	U	4,0
I	7,6	V	4,7
J	2,1	W	0,0
K	3,7	X	0,1
L	3,8	Y	2,9
M	3,2	Z	3,1

Po sestavení tabulky vah písmen bylo třeba ještě rozhodnout, která gesta jsou jednodušší a tudíž by více vyhovovala častějším písmenům. Nakonec byla gesta z různých hledisek rozdělena na vertikální, horizontální, jedním prstem, dvěma prsty (zde další dělení na stejný nebo opačný směr pohybu) a také na gesta v jedné nebo v obou polovinách plochy.

Jako nejjednodušší gesta byla vybrána svislá gesta jedním prstem, ta byla použita pro nejčastější znaky. Další nejčastější znaky v pořadí byly přiřazeny vodorovným gestům jedním prstem a tak dále, až byla namapována celá abeceda s tím, že na nejméně pohodlné nebo nejhůře proveditelné pohyby zbyla písmena s minimální frekvencí výskytu (X, Q a podobně).

Vzhledem k tomu, že se tento layout nepodobá žádnému dosud používanému, dá se předpokládat horší intuitivnost a horší možnost rychle se psaní na multitouch klávesnici naučit, nicméně v dlouhodobém měřítku bude toto frekvenční rozložení výhodnější než alfabetské. Kromě toho zde můžeme vzít v potaz cílovou skupinu mladých lidí, kteří obvykle nemají problém se podobné nové věci rychle naučit.

### **3.4.3 Stručný princip – finální aplikace**

Hlavní princip zadání znaků zůstal od prototypu zachován – rozhodující faktory v určení znaku jsou tedy směr tažení, bod začátku pohybu, bod jeho konce a počet prstů. Z důvodu, aby metoda nevyžadovala přílišnou přesnost a aby bylo možné vytvořit spolehlivější rozpoznání gest, nehraje žádnou roli tvar ani rychlost gesta.

Zadávací metoda v základním tvaru obsahuje písmena a několik dalších funkcí potřebných pro běžné psaní (přepínání velkých / malých písmen, tečka, čárka, mezera, ...). Pro čísla a speciální znaky má uživatel možnost přepínat mezi třemi módy klávesnice. Přepínání probíhá speciálním gestem, u kterého bylo cílem, aby nešlo snadno zaměnit za jiná gesta a nedocházelo tak k nechtěným přepnutím klávesnice (toto gesto je tažení dvěma prsty přes dělicí čáru uprostřed plochy). Druhý mód tedy obsahuje čísla a několik dalších častěji používaných speciálních znaků, zatímco třetí, poslední mód obsahuje čistě symboly, které se v textu příliš často nevyskytují. Všechny tři módy můžeme vidět na obrázku 12, kde je zároveň vidět, že gesta potřebná pro všechny druhy psaní mají funkci zachovánu (mezera, mazání).



Obrázek 12 - Módy klávesnice

Přepínání funguje cyklicky jedním směrem (uživatel tedy může přepínat z prvního na druhý, z druhého na třetí, z třetího na první a takto stále dokola). Pro případy některých speciálních zadávacích polí (například pro telefonní číslo) klávesnice automaticky zvolí mód, který obsahuje čísla a znaky použitelné u telefonních čísel, tedy mód číslo 2.

## 4 Implementace

V této kapitole se budeme věnovat zejména principu rozpoznání gest, jakým způsobem je s ním pracováno. Budeme se také věnovat tomu, jaké prostředky byly použity pro implementaci a velmi hrubě načrtneme strukturu tříd a jejich funkcí.

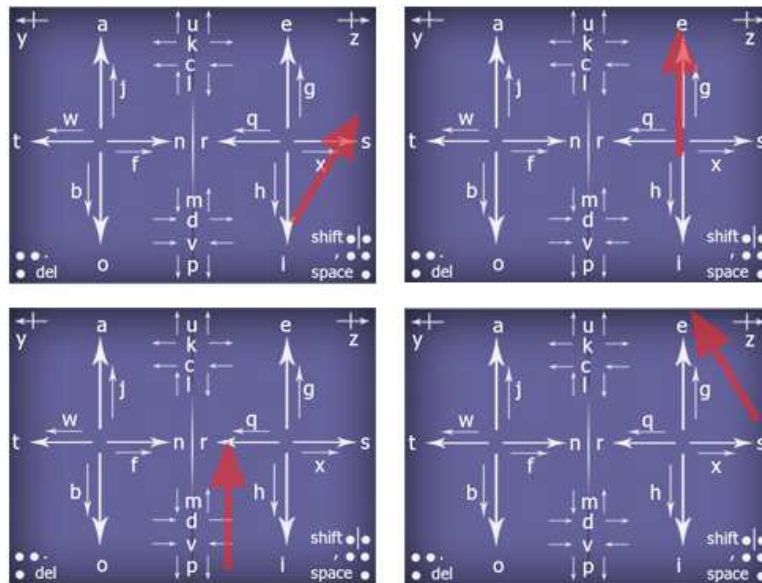
### 4.1 Způsob rozpoznání gest

V průběhu vývoje aplikace se způsob rozpoznání gest poměrně významně změnil. Zpočátku byla snaha, aby rozpoznávání probíhalo již při zadávání gesta postupně, porovnáváním jednotlivých souřadnic, kterými prsty v průběhu zadávání gesta prošly. Toto by ve výsledku umožňovalo pracovat i s tvary gest, nicméně to se postupem času ukázalo jako nepotřebné, bylo proto přistoupeno k jinému, jednoduššímu principu rozpoznávání.

V současné aplikaci funguje rozpoznávání gest zcela jednoduše, na základě sestavování vektorů, které nám umožňují snadno dopočítat úhel pohybu i velikost vektoru. Při dotyku na zadávací ploše si aplikace zaznamená souřadnice bodu, kde pohyb začal. Uživatel pak pohybuje prstem směrem odpovídajícím písmenu, které si přeje napsat. Další bod si aplikace zaznamená, ve chvíli kdy dotyk skončí. Máme tedy souřadnice dvou bodů, což nám umožňuje sestavit vektor. Veškeré pohyby provedené mezi začátkem a koncem gesta tím pádem nehrají roli, protože se zaznamenává jen začátek a konec.

Ve chvíli, kdy máme takto sestaven vektor, aplikace spočítá jeho velikost. To nám umožní rozhodnout, zda se jednalo o poklepnutí na displej (což je gesto, které je přiřazeno například mezeře nebo mazání znaků), nebo zda se jednalo o pohyb nějakým směrem. K nějaké změně souřadnic dochází vždy, je proto třeba určit hraniční velikost vektoru, které ještě nebude považována za pohyb jakýmkoliv směrem. Pokud je velikost vektoru menší než tato velikost, bude se výsledné gesto považovat za poklepnutí na displej. Pokud je větší, bude se výsledné gesto považovat za pohyb určitým směrem.

Ve chvíli, kdy víme, že se nejedná pouze o poklepnutí, program z vektoru dopočítá pomocí goniometrických funkcí ještě jeho úhel. Když známe úhel, můžeme určit směr vektoru.



Obrázek 13 - Příklad - různé možnosti pro napsání písmene E

Gestům jsou vyhrazeny úhly s rozsahem 90 stupňů, což nám zajišťuje poměrně velkou odolnost proti nechtěnému zadání jiného znaku (jak je vidět na obrázku 13). Ve chvíli, kdy známe velikost i úhel vektoru, ještě potřebujeme vědět, v jaké části zadávací plochy se pohyb udál a zda nezasahoval do obou částí. To zjistíme jednoduše tak, že porovnáme souřadnice obou bodů na ose X. Pokud jsou obě souřadnice menší, než je polovina šířky displeje, víme, že se jednalo o pohyb v levé polovině zadávací plochy. Pokud jsou naopak větší, víme, že pohyb se udál vpravo. Pokud je jedna souřadnice vlevo a jedna vpravo, můžeme podle jejich pořadí určit, zda gesto dělicí čáru překračuje zleva doprava či zprava doleva.

U multitouch gest je situace zcela obdobná, protože aplikace si uchovává data pro každý dotyk zvlášť. Pohyby jednotlivých prstů se proto vyhodnotí nezávisle na sobě a až následně se z jejich kombinace určí, o které gesto a tedy i o který znak se má jednat. Je proto možné používat nejen gesta, při kterých se oba prsty pohybují stejným směrem, ale i taková, kde mají směr opačný, nebo obecněji řečeno rozdílný. Totéž platí i pro rozdělení zadávací plochy, které taktéž pro multitouch gesta není překážkou (u každého dotyku zvlášť poznáme, ve které polovině plochy proběhl).

## 4.2 Implementace pro OS Android

Aplikace byla pro OS Android vyvinuta jako metoda vstupu textu, což je v rámci Androidu systémová služba (nejedná se tedy doslova o aplikaci). K tomu bylo třeba využít třídu `InputMethodService`, která slouží k tomu, aby nám umožnila tyto vstupní metody

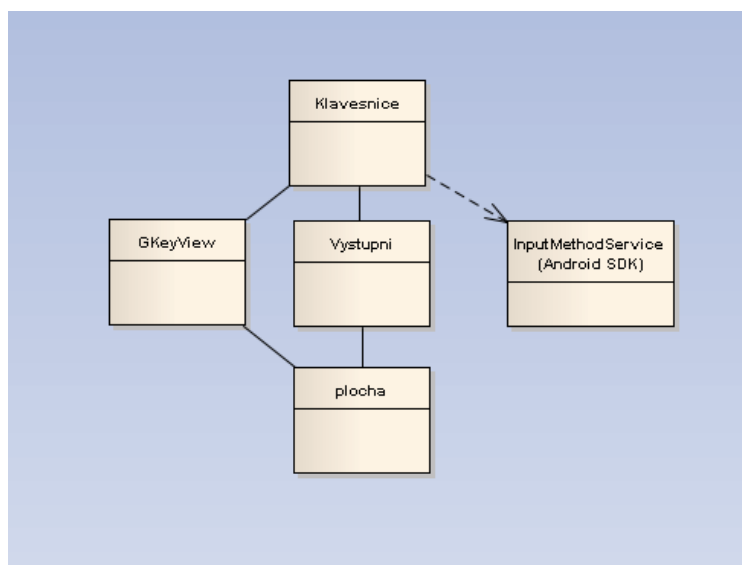


vytvářet. Hlavní třída naší aplikace, `Klavesnice.java`, proto od této třídy dědí. Tato třída se stará o to, aby se metoda objevila v seznamu vstupních metod na zařízení a dále má na starosti komunikaci aplikace se zbytkem prostředí (zde se rozpoznává, jaký druh vstupního pole je zrovna aktivní, posílají se zde znaky na výstupní stream atd.).

Dále aplikace obsahuje třídu `Vystupni.java`, která se stará o posílání znaků do hlavní třídy na základě provedeného gesta.

Nejdůležitější třídou aplikace je `plocha.java`, kde probíhá veškeré rozpoznávání a detekování gest. Vznik vektorů v aplikaci nám umožňuje vyhodnocování jednotlivých objektů `MotionEvent`, které jsou nám posílány přímo od vstupní plochy.

Poslední třídou je `GKeyView.java`, ta se stará o zobrazení aplikace a o správné přiřazení popisků na jednotlivé pozice. V případě změny rozložení nebo módu klávesnice má za úkol updatovat popisky vždy tak, aby odpovídaly tomu, co skutečně v daném směru pohybu můžeme napsat. Na obrázku 14 můžeme vidět zjednodušený diagram tříd aplikace pro představu její struktury.



Obrázek 14 - Zjednodušený diagram tříd

## 5 Dlouhodobá případová studie

Pro získání informací o tom, zda bude ve výsledku vstupní metoda vytvořená v této práci využitelná v reálném prostředí a hlavně reálnými uživateli, bylo provedeno dlouhodobější testování s uživateli z cílové skupiny. Za účelem co možná nejširšího prozkoušení a největšího množství připomínek byli vybráni participanti s různou mírou zkušenosti s dotykovými zařízeními – od těch, kteří v podstatě nikdy dotykový telefon ke vstupu textu nepoužívali, přes ty, kteří jej vlastní ale používají některou z běžných metod vstupu (např. dotykovou QWERTY klávesnici) až po ty, kteří pro vstup textu používají méně tradiční, nicméně pro psaní na dotykovém zařízení vhodnější metody (např. GO Keyboard – obdoba Swype).

V této kapitole se budeme věnovat tomu, jak testování probíhalo, co k němu bylo použito, jaké byly profily participantů testu a zejména tomu, jaké byly jeho výsledky, které by mohly ovlivnit budoucí práci na metodě.

### 5.1 Popis experimentu

Experiment měl za úkol ověřit možnosti vstupní metody v konfrontaci s ostatními, již používanými metodami vstupu textu. Nebyla proto testována jen metoda vyvinutá v této práci, ale u každého participanta byla zároveň testována i jeho vstupní metoda, na kterou je zvyklý (pro participanty, jejichž zkušenosti neobsahovaly práci s dotykovými telefony, byla zvolena jako nativní metoda dotyková QWERTY klávesnice). To nám ve výsledku umožňuje porovnávat novou metodu v konfrontaci se zažitými a naučenými metodami. Pro lepší srovnání by nicméně bylo nutné, aby uživatelé používali multitouch klávesnici i pro psaní v běžném životě, bez návaznosti na testovací sezení – pro naučení zcela nové metody do té míry, aby se její rychlost a další faktory daly srovnávat s ostatními, totiž nestačí jen několik sezení při testech.

#### 5.1.1 Způsob testování

K testům bylo přizváno dohromady 5 uživatelů, kteří měli (jak již bylo řečeno výše) různou míru zkušenosti s dotykovými displeji. To nám umožnilo získat představu o tom, o kolik lépe

nebo hůře se metodu bude učit zcela nový uživatel a naopak o kolik lépe nebo hůře se ji naučí uživatel vlastníci a používající dotykový telefon.

S každým uživatelem proběhlo dohromady 7 sezení (sessions). V každém z těchto sezení měl uživatel za úkol vykonat dva testy – jeden test s naší metodou a jeden test s vlastní metodou, která mu ke psaní nejvíce vyhovuje.

Testy měly formu opisování náhodně vybraných krátkých vět. Pro každou větu byl zaznamenán počet znaků, které obsahuje, počet uživatelových akcí (to nám umožnilo zjistit, jak často uživatel znaky nebo slova maže a opravuje) a počet neopravených chyb, s nimiž uživatel napsání věty potvrdil.

Doba každého testu byla 10 minut, počet vět k opsání nebyl dopředu omezen (z důvodu, abychom mohli u uživatelů schopných psát rychleji testovat po celých 10 minut). V jednom sezení tedy uživatel vykonal 2 desetiminutové testy, z nichž jeden byl s naší metodou a jeden s metodou podle vlastního výběru. Pořadí metod bylo při každém testování náhodně určeno, abychom zamezili případnému efektu učení.

Při testech byla použita pomocná aplikace, která byla vytvořena přímo pro účely tohoto testování a bude blíže popsána níže v části věnované setupu testů. Ta měla za úkol odměřit vždy 10 minut času, spočítat počty znaků a vět a kromě toho také náhodný výběr vět z předem dané množiny.

Před prvním testem byl každý uživatel podrobně seznámen s tím, jak vstupní metoda funguje a jak má zacházet s pomocnou testovací aplikací. Kromě toho byly se všemi uživateli první testy provedeny s asistencí, mohli se tedy kdykoliv zeptat na jakékoliv nejasnosti.

Všichni uživatelé měli v případě zájmu možnost si nechat tuto vstupní metodu nainstalovat na vlastní telefon (pokud tu možnost podporoval). V takovém případě mohli zkoušet metodu používat i k běžnému psaní. U uživatelů, kteří si nechali na vlastní telefon metodu nainstalovat a metodu používali i mimo testování, to bude uvedeno v přehledu participantů.

## 5.2 Cíle testování

Data získaná testováním nám umožnila získat dostatek podkladů k tomu, abychom mohli spočítat hodnoty KSPC a WPM pro každého uživatele a pro jednotlivá sezení. Z těchto hodnot bylo možno sestavit i učící křivku (vzhledem k počtu sezení poměrně hrubou). Kromě toho bylo po provedení všech testů s každým uživatelem konzultováno za účelem získat konkrétní připomínky k funkci nebo principu metody.

## 5.3 Setup

Pro testování byl stejně jako v předběžné studii ve většině případů použit mobilní telefon LG Optimus Black (P970) s operačním systémem Android verze 2.2. Vzhledem k použití pomocné aplikace nicméně měli uživatelé možnost testovat své nativní metody i na vlastní pěst na jiných telefonech s operačním systémem Android. Vždy bylo nicméně upřednostňováno, aby uživatel testoval s asistencí, aby bylo možno na místě řešit případné nefunkčnosti nebo nejasnosti ohledně multitouch klávesnice.

Kromě mobilního telefonu s nainstalovanou vstupní metodou a pomocnou aplikací pro testování nebylo třeba žádných dalších speciálních pomůcek.

### 5.3.1 Pomocná aplikace

Pomocná aplikace byla realizována jako jednoduchý program pro telefony s OS Android. Při implementaci nebyly použity žádné prvky z novějších verzí Android SDK, aby aplikace mohla fungovat i na starších dotykových telefonech.

Ovládání aplikace je velice jednoduché, aby se uživatelé kromě nové vstupní metody nemuseli zabývat ještě tím, jak pracovat s testovací aplikací. Je realizováno 4 tlačítka, z nichž první (Zacít) slouží ke spuštění testu, druhé (Dalsi veta) slouží k potvrzení zadání věty a načtení další náhodné věty, třetí (Vyhodnotit) slouží k zobrazení statistik po dokončení testu a poslední tlačítko slouží k resetu aplikace, aby mohl být proveden další test.

Kromě 4 ovládacích tlačítek aplikace obsahuje informaci o tom, kolik do konce testu zbývá času. Pod tímto údajem se vždy zobrazuje věta, kterou má uživatel přepsat. Poslední částí aplikace je vstupní textové pole, kam má za úkol zapisovat přečtené věty. Po napsání a potvrzení věty se toto textové pole vždy automaticky vyčistí, aby uživatel nemusel ztrácet při testu čas s mazáním předchozího textu.

Na obrázku 15 je možno vidět spuštěnou pomocnou aplikaci v průběhu testu s multitouch klávesnicí.



Obrázek 15 - ukázka pomocné testovací aplikace

Typický průběh testu v aplikaci vypadá tak, že uživatel nejprve spustí aplikaci, poklepe prstem do textového pole (aby na obrazovce vyvolal klávesnici – ať už svou vlastní nebo multitouch keyboard). Poté stiskne tlačítko **Zacit**, čímž začne odpočítávání 10 minut času vyhrazeného pro test a zobrazí se první věta k opsání. Uživatel opíše větu a stiskne tlačítko **Dalsi veta**. Tím se v aplikaci porovná, to co uživatel napsal s větou, kterou měl opisovat a údaje se uloží. Zároveň se smaže obsah textového pole a zobrazí se další věta k opsání. Takto test pokračuje, dokud čas nedejde a na ukazateli času není údaj 0:0. V tu chvíli už program uživateli nedovolí potvrzovat další věty. Pro zobrazení výsledků po konci testu uživatel klikne na tlačítko **Vyhodnotit**, načež se zobrazí několik údajů, které jsou v testu sledovány. Po konci testu je ještě možnost stisknout tlačítko **Novy test** a tím vynulovat všechny údaje a připravit nový test. Aplikace je naprogramována tak, aby ignorovala rozdíly mezi velkými a malými písmeny a také aby znaky z diakritikou považovala za téže znaky bez diakritiky. Je to tak z toho důvodu, aby nebyly v testech znevýhodněny metody s predikcí (ty mají často ve slovnících slova s diakritikou a taková, která obsahují velká písmena).

Výstupem aplikace po stisknutí tlačítka **Vyhodnotit** je několik čísel – první z nich je počet napsaných vět, druhé z nich počet napsaných znaků (to potřebujeme k určení WPM), další je počet chyb, s nimiž uživatel věty potvrdil (pro určení náchylnosti jednotlivých metod k chybám) a poslední číslo je počet akcí, tedy počet stisků tlačítek nebo provedených gest

(to nám naopak umožní v porovnání s počtem napsaných znaků určit KSPC jednotlivých metod).

## 5.4 Přehled participantů

Vzhledem ke snaze vybrat uživatele, kteří nemají přílišné zkušenosti s dotykovými displeji i uživatele kteří jsou na ně zvyklí, bylo nutné se v několika případech mírně odchýlit od cílové skupiny (netechnicky vzdělaní mladí lidé). Všichni, kdo se účastnili testování, spadají do kategorie mladých lidí (všichni studenti vysokých škol), nicméně někteří právě absolvují technické školy (takoví lidé však zpravidla nejvíce používají dotykové smartphony, což pro nás znamená různorodější spektrum uživatelů). U uživatelů, kteří se pokoušeli multitouch klávesnici používat i mimo testy, to vždy je výslovně uvedeno (aby bylo možné sledovat, o kolik lépe si pak v testech vedou).

V popisu každého participanta se objeví věk, typ vzdělání, pohlaví, popis zkušeností s dotykovými displeji, údaj, jaký druh telefonu vlastní a také jakou používá metodu vstupu textu. Kromě toho se v popisech mohou objevit další relevantní informace, které mohou souviset se schopnostmi participanta učit se nové metodě.

### 5.4.1 Participant 1

Muž, student technické školy, 22 let. Vlastní dotykový telefon s operačním systémem Android. Zařízení s dotykovým displejem běžně používá, má zkušenosti s několika metodami vstupu textu, nejčastěji používá GO Keyboard (což je obdoba Swype, metody, která je popsána dříve v této práci). Po seznámení s metodou si nechal na svůj telefon nainstalovat i multitouch klávesnici, kterou se alespoň občas snažil používat pro běžné psaní.

Psaní na telefonu běžně používá, píše nejen SMS zprávy, ale i e-maily, příspěvky do diskusí a podobně.

### 5.4.2 Participant 2

Žena, studentka umělecké školy, 20 let. Vlastní klasický telefon značky Nokia, s dotykovými displeji nemá v podstatě žádné zkušenosti. Jako vlastní metodu pro psaní na mobilních

telefonech používá T9 a někdy MultiTap (obě metody viz přehled v kapitole Vstup textu na klasických telefonech). Běžně píše na počítači, proto pro testování používá dotykovou QWERTY klávesnici.

Na mobilním telefonu používá vstup textu primárně pro psaní SMS zpráv, občas také pro poznámky a podobně. Rozsáhlejší texty na telefonu nepíše.

### **5.4.3 Participant 3**

Muž, student technické školy, 22 let. Vlastní dotykový telefon s Windows Phone 7, takže je na zařízení s dotykovým displejem zvyklý. Používá dotykovou QWERTY klávesnici s predikcí slov.

Vstup textu na telefonu používá zejména pro SMS zprávy, někdy také kratší e-maily.

### **5.4.4 Participant 4**

Muž, student umělecké školy, 20 let. Vlastní klasický telefon značky Nokia, nicméně má i určitou zkušenost s dotykovými telefony. Používá MultiTap, na dotykových telefonech má zkušenosti s dotykovou QWERTY klávesnicí, kterou používá i jako druhou metodu při testování.

Vstup textu na telefonu používá pouze pro psaní SMS zpráv, nicméně píše jich velké množství.

### **5.4.5 Participant 5**

Žena, studentka školy s humanitním zaměřením, 21 let. Vlastní dotykový telefon s OS Android, v ovládání telefonu s dotykovým displejem tedy disponuje určitými zkušenostmi. Jako vstupní textovou metodu používá QWERTY s predikcí slov.

Vstup textu na svém telefonu používá primárně na e-maily, občas i pro jiné internetové služby a také SMS zprávy.

## 5.5 Výsledky testů

Po každém testu a zaznamenání čísel, které z něho vyšly, bylo možno dopočítat hodnoty KSPC a WPM, které byly následně zaneseny do tabulky, ze které byly po provedení všech testů vytvořeny i učící křivky. Pro každého uživatele bylo toto provedeno zvlášť, aby bylo možné porovnat učení metody v jeho vlastním kontextu.

Níže bude nejprve následovat rozpis získaných dat pro jednotlivé participanty. Pro reálnější posouzení možností metody by bylo třeba ještě delšího experimentu (výrazně více než 7 sezení), případně testovat s takovými uživateli, kteří by metodu používali v reálném životě a nejen při testech (což se v našem případě podařilo u jednoho uživatele). Pro hrubou představu byla u každého participanta u obou metod provedena regresní analýza učení a vytvořena předpokládaná učící křivka na dalších 20 sezení dopředu. Jedná se ovšem pouze o odhad, reálná učící křivka by s největší pravděpodobností vypadala jinak.

Ve finálním přehledu budou také uvedeny nejčastější postřehy a připomínky samotných uživatelů, z nichž některé by měly být zohledněny v případné budoucí práci na aplikaci.

Řádek „chyby“ v tabulkách představuje počet neopravených chyb. Opravené chyby jsou zahrnuty ve výpočtu KSPC (je započítán celkový počet stisků tlačítek nebo vykonaných gest v poměru k zadanému textu k opsání).

### 5.5.1 Participant 1

Metodu velmi rychle pochopil a projevil i zájem o její používání mimo testy. Jak je vidět z tabulky pro multitouch klávesnici, promítlo se to také do výsledných měřených hodnot. Vzhledem ke zvyku na metodu podobnou Swype mu nedělalo problém pro psaní používat gesta namísto stisků virtuálních tlačítek.

Tabulka 2 - participant 1, multitouch

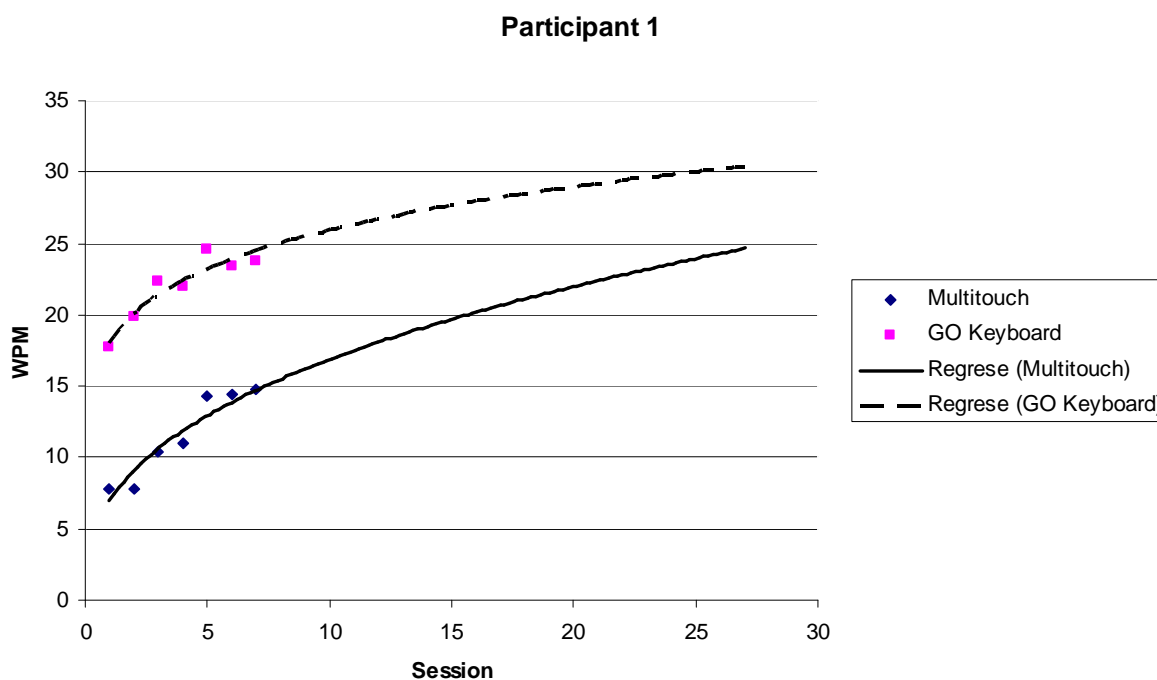
	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7
KSPC	1,3257	1,2519	1,2703	1,2646	1,1704	1,1740	1,1863
WPM	7,86	7,78	10,36	10,96	14,32	14,48	14,82
chyby	106	2	1	0	1	1	0



Tabulka 3 - participant 1, GO Keyboard

	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7
KSPC	0,6156	0,6317	0,5179	0,4973	0,5102	0,5449	0,5407
WPM	17,74	19,82	22,32	21,96	24,62	23,38	23,74
chyby	0	10	3	2	2	11	7

Jak je vidět z tabulek výše, u obou vstupních metod došlo za dobu testování k dramatickému nárůstu rychlosti textového vstupu. Zároveň u obou metod převládá nízká chybovost (s výjimkou prvního testu multitouch klávesnice, kdy se uživatel s metodou seznamoval a zároveň se snažil věty zadávat rychle). Totéž se týká i počtu opravených chyb, jehož pokles se nám promítá do postupně klesajícího KSPC. Velmi nízké hodnoty KSPC u GO Keyboard jsou dány především tím, že se jedná o prediktivní metodu, která je kombinována se stejným principem psaní jako Swype (popsáno v přehledu metod v dříve v této práci), tedy táhlé pohyby po klávesnici. To nám sice dává menší hodnoty KSPC, nicméně je třeba si uvědomit, že jednotlivé pohyby trvají déle, WPM zde proto není vyšší stejnou měrou, jakou je nižší KSPC.



Graf 1 - odhad učících křivek pro participanta 1

V grafu 1 můžeme vidět, že u tohoto participanta by nejspíše multitouch metoda nebyla rychlejší ani po 20 dalších sezeních (dá se nicméně předpokládat, že hodnoty WPM by u GO

Keyboard už nerostly tak rychle, jak vyšlo z regrese). Pravděpodobnější výsledky odhadů pomocí regrese by taktéž přinesl větší počet sezení.

### 5.5.2 Participant 2

Vzhledem k malé zkušenosti s dotykovými displeji a celkově s méně tradičními metodami vstupu textu zde seznamování s metodou trvalo o něco déle. Kromě testovacích sezení multitouch klávesnici nepoužívala.

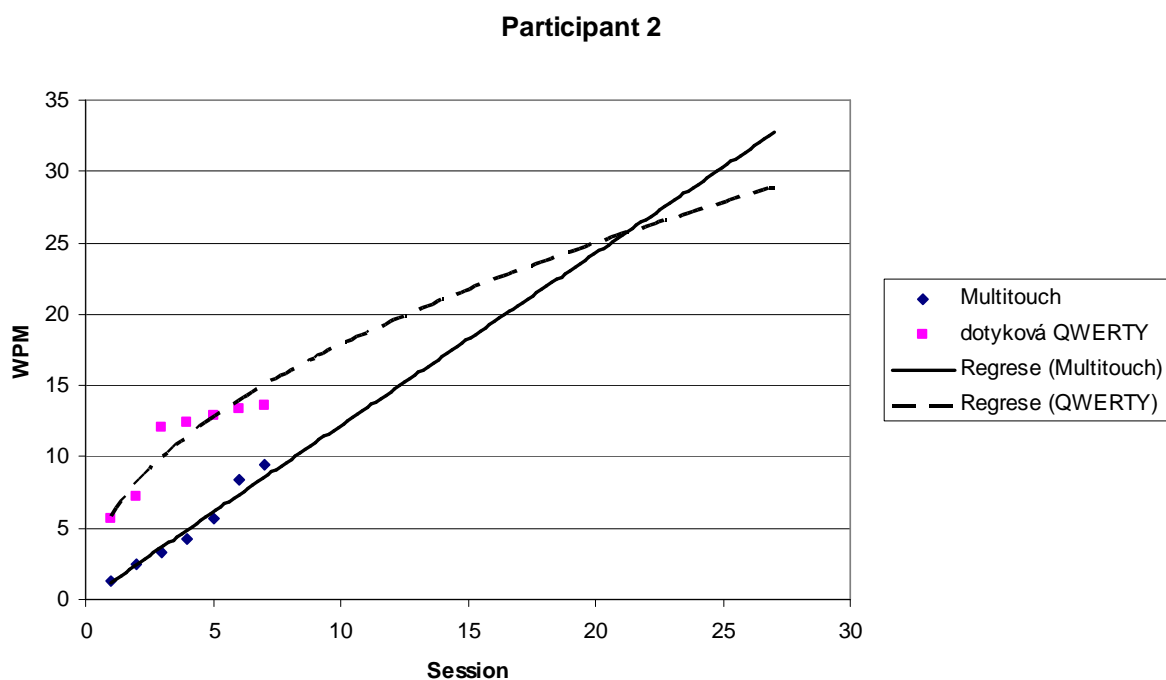
**Tabulka 4 - participant 2, multitouch**

	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7
KSPC	2,1642	1,6000	1,7500	1,7143	1,5385	1,4426	1,4650
WPM	1,34	2,50	3,36	4,20	5,72	8,36	9,42
chyby	2	20	0	4	3	7	5

**Tabulka 5 - participant 2, dotyková QWERTY**

	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7
KSPC	1,3754	1,4417	1,1728	1,1846	1,1975	1,1833	1,1789
WPM	5,70	7,20	12,04	12,46	12,86	13,42	13,64
chyby	0	3	6	9	1	8	2

U tohoto participanta se jednoznačně promítla předchozí nezkušenost s dotykovými displeji, protože počáteční sezení u obou metod mají markantně horší výsledky, než sezení na konci testování. Jak je vidět, vyšší hodnoty WPM a nižší KSPC se mnohem dříve objevily u dotykové QWERTY klávesnice, můžeme proto konstatovat, že pro tohoto participanta by bylo učení multitouch klávesnice pomalejší.



**Graf 2 - odhad učících křivek pro participanta 2**

Jak můžeme vidět na grafu 2, pro tohoto participanta by podle odhadu regresí měly obě metody mít stejnou rychlost z hlediska WPM po 20. sezení. Dá se tedy předpokládat, že by se u tohoto participanta mohly obě metody časem dostat na stejnou úroveň rychlosti.

### 5.5.3 Participant 3

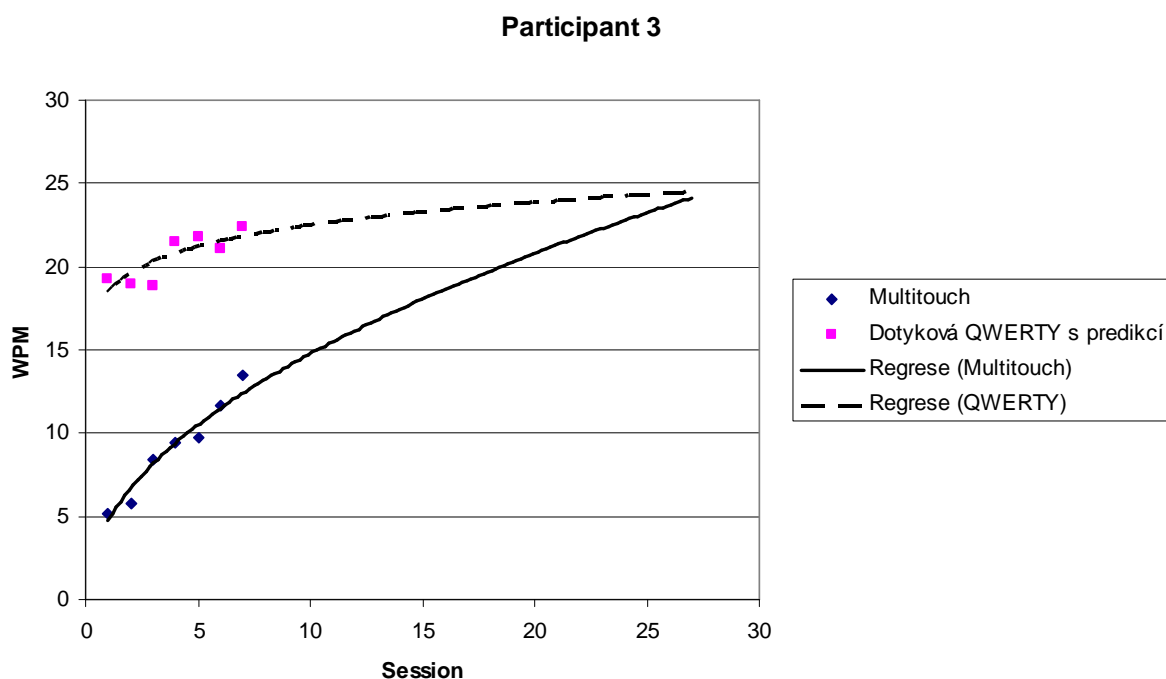
**Tabulka 6 - participant 3, multitouch**

	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7
KSPC	1,5247	1,4754	1,4461	1,4827	1,3651	1,3442	1,2898
WPM	5,16	5,82	8,44	9,46	9,68	11,62	13,44
chyby	24	7	9	2	13	7	10

**Tabulka 7 - participant 3, dotyková QWERTY s predikcí**

	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7
KSPC	0,8552	0,8317	0,8587	0,8116	0,8329	0,7978	0,8601
WPM	19,26	18,98	18,86	21,46	21,78	21,12	22,44
chyby	0	0	0	0	0	0	0

Jak je vidět, tento participant zvládal svoji metodu na velmi dobré úrovni a stejně tak zvládal učení multitouch klávesnice. Nárůst WPM u vlastní metody participanta opět připisují schopnosti učení u jeho prediktivní vstupní metody.



Graf 3 - odhad učících křivek pro participanta 3

Jak je vidět z grafu 3, pro třetího participanta by nejspíše taktéž obě metody dosáhly stejné rychlosti, bylo by to nicméně později než u participanta 2. V rámci našeho testu a odhadu na dalších 20 sezení nicméně zůstává dotyková QWERTY klávesnice s predikcí jako rychlejší metoda vstupu textu.

#### 5.5.4 Participant 4

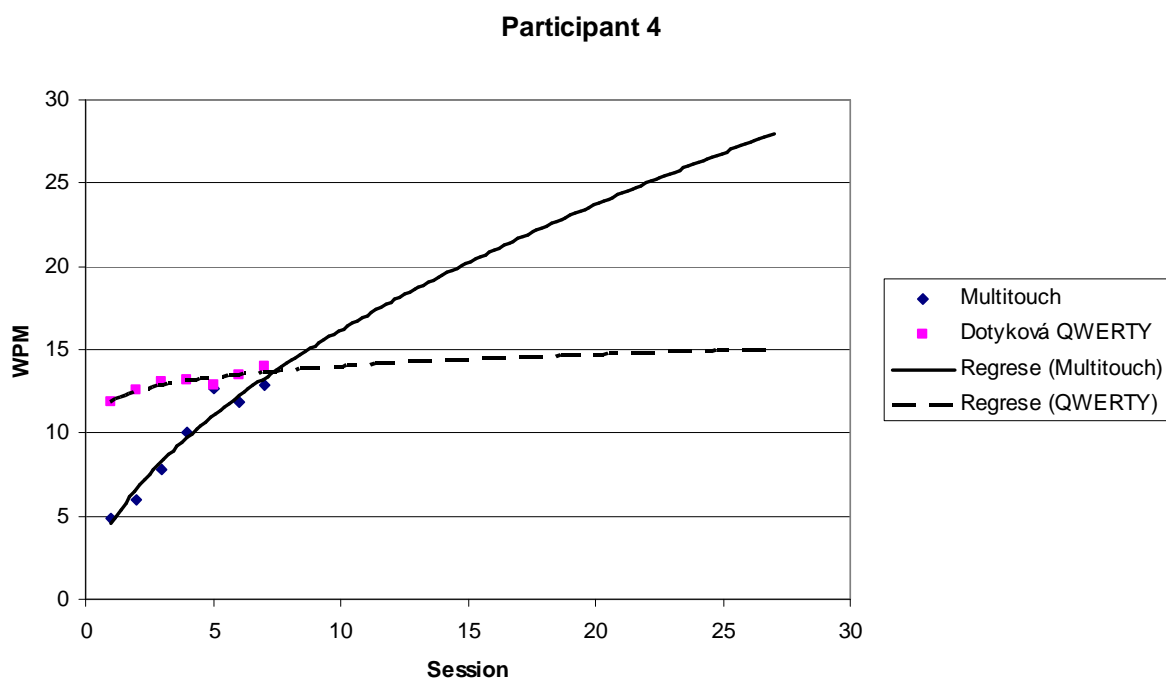
Tabulka 8 - participant 4, multitouch

	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7
KSPC	1,1276	1,1571	1,1644	1,1672	1,1611	1,1824	1,1799
WPM	4,86	5,98	7,78	10,04	12,64	11,86	12,88
chyby	0	0	1	1	1	0	1

Tabulka 9 - participant 4, dotyková QWERTY

	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7
KSPC	1,2441	1,1148	1,1086	1,0897	1,1112	1,1243	1,0942
WPM	11,88	12,54	13,06	13,14	12,84	13,46	14,02
chyby	2	1	0	1	0	3	0

Z tabulek výše vidíme, že u obou metod se objevují velmi dobré hodnoty KSPC (uživatel dal přednost psaní bez chyb před rychlostí). Přesto se u tohoto uživatele v posledních sezeních podařilo docílit téměř stejné hodnoty WPM u obou metod (přičemž hodnota WPM u dotykové QWERTY rostla poměrně pomalu, pro multitouch je nárůst výraznější). Z toho vyplývá teze, že pro pečlivější druh psaní by mohla multitouch metoda být vhodnější (zřejmě proto, že nevyžaduje takovou přesnost při ovládání a ve chvíli naučení není tolik náchylná k chybám).



Graf 4 - odhad učicích křivek pro participanta 4

Jak můžeme vidět z grafu 4, pro tohoto participanta by pravděpodobně byla multitouch metoda oproti dotykové QWERTY klávesnici bez zvláštních funkcí rychlejší už v 8. sezení. Výrazný nárůst rychlosti psaní multitouch metodou v dalších sezeních je v této míře nepravděpodobný, nicméně pro tohoto participanta se dá předpokládat, že v dalších sezeních by s multitouch klávesnicí dosahoval lepších výsledků než s dotykovou QWERTY.

### 5.5.5 Participant 5

Tabulka 10 - participant 5, multitouch

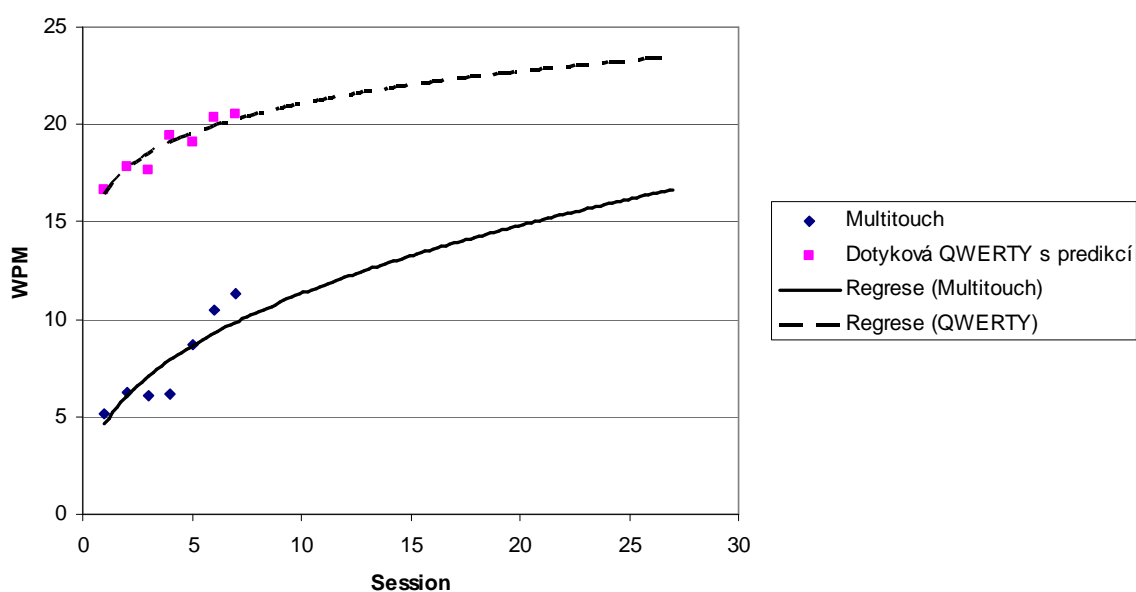
	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7
KSPC	1,7245	1,6577	1,4534	1,3665	1,3471	1,3822	1,2297
WPM	5,16	6,22	6,06	6,14	8,74	10,46	11,32
chyby	23	11	13	9	17	8	2

Tabulka 11 - participant 5, dotyková QWERTY s predikcí

	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6	Session 7
KSPC	1,0325	0,9452	0,9587	0,8856	0,9023	0,8777	0,8803
WPM	16,6	17,82	17,66	19,46	19,08	20,32	20,50
chyby	8	5	0	0	0	0	3

U tohoto participanta opět můžeme vidět rapidní nárůst WPM a pokles KSPC u naší metody, nicméně dosti výrazný nárůst rychlosti vidíme i u jeho nativní metody. To je s největší pravděpodobností dáno tím, že prediktivní metody se učí slova a přizpůsobují jejich prioritu frekvenci jejich používání. Po prvních testech se frekvence slov v testovacích větách zvětšila a predikce probíhala na základě menšího počtu znaků.

Participant 5



Graf 5 - odhad učicích křivek pro participanta 5

Z grafu 5 vidíme, že pro tohoto participanta by pravděpodobně zůstala výhodnější jeho nativní metoda, tedy dotyková QWERTY klávesnice s predikcí slov. Vykazuje nicméně i výrazný nárůst rychlosti psaní u naší multitouch metody.

### 5.5.6 Postřehy a připomínky participantů

Mezi nejčastější připomínky patřilo zejména to, že by klávesnice měla nabízet lepší zpětnou vazbu a hlavně umožnit uživateli, aby si tuto zpětnou vazbu sám nastavil. Některým uživatelům vibrace nevyhovovala, jiným nevyhovoval zvuk. Mezi další připomínky patřily výhrady k mapování znaků, v tomto ohledu se nicméně nestalo, že by více uživatelů uvedlo stejný znak jako nevhodně namapovaný, dá se tedy předpokládat, že se jedná spíše o osobní preference a o to, že pro každého uživatele je intuitivnější jiné rozložení.

## 5.6 Diskuse

Z výsledků testování vyplývá jedna věc – učení nové multitouch metody je rozhodně snazší pro uživatele, kteří jsou zvyklí používat dotykové displeje. Zároveň je vidět, že učení metody je pravděpodobně poněkud náročnější, protože uživatel, který metodu používal i mimo testování (když měl příležitost) dosáhl výrazně lepších hodnot WPM a KSPC než ti, kteří se metodu učili pouze v době testování.

Přestože v žádném z testů nová metoda neobstála jako rychlejší, můžeme se na základě odhadů dalšího vývoje regresí (grafy v přehledu výše) domnívat, že alespoň u některých uživatelů by se naše metoda po dalších sezeních stala rychlejší (zejména u těch, kteří jako nativní metodu měli méně sofistikované vstupní metody – tedy například dotykovou QWERTY klávesnici bez zvláštních funkcí). Uživatelé zpravidla po testování vyzdvihovali pohodlnější zadání textu, než například na QWERTY klávesnici. Kromě toho lze předpokládat, že delší učení metody by přineslo zdaleka jiné hodnoty KSPC a WPM. Uživatelé totiž již znají layout QWERTY klávesnice, zatímco u multitouch klávesnice se kromě učení způsobu zadávání gesty uživatel musí sít s rozložením znaků na obrazovce tak, aby je při psaní nemusel příliš dlouho hledat. V úvahu můžeme vzít i fakt, že k plnému naučení vstupní metody většina uživatelů potřebuje 100 a více hodin používání, což je řádově více v porovnání se 70 minutami, které někteří participanta strávili s naší metodou při testování.

Z obecného pohledu jsou hodnoty WPM, které v mnoha konečných sezeních přesahovaly 12 WPM, srovnatelné s dotykovou QWERTY klávesnicí bez speciálních funkcí (pokud bereme v potaz běžného uživatele, pro expertního uživatele se hodnoty QWERTY pohybují mnohem výše).

Z hlediska chybovosti zde nelze najít výrazný problém, pokud pomineme extrémní u jednotlivých participantů (tyto extrémní se týkají zejména prvních sezení). Velký rozsah úhlů pro jednotlivá gesta napomáhá tomu, aby se vyskytlo minimum chybně zadaných znaků (v případě opravených i neopravených chyb s multitouch klávesnicí všichni participanté uvedli spíše to, že byly způsobeny jejich nedostatečným zvykem na rozložení znaků).

Pro přesnější zjištění vlastností multitouch klávesnice by proto byla potřeba ještě dlouhodobější studie s tím, že participanté takového experimentu by metodu používali v reálném životě s tím, že v jednotlivých testech by se zjistilo, jak rychle probíhá učení, pokud nejsme při psaní omezeni jen na časově omezené úseky.



## 6 Závěr

Při vývoji této metody bylo zpočátku cílem vyvinout metodu s rychlejším zadáním textu. Větší úsilí však bylo nakonec věnováno celkovému principu zadání znaku a také tomu, aby se jednalo o metodu, která nebude vyžadovat takovou přesnost zadání. Vzhledem k faktu, že oproti jiným vstupním metodám zde uživatel musí správně určit pouze směr a jednu (nebo obě) ze dvou ploch, je požadavek na přesnost výrazně menší, než například u klávesnice, která má na displeji přenosného zařízení přes 30 velmi malých tlačítek, která je poté obtížné správně stisknout. Podle reakcí účastníků po testování můžeme říci, že snížit požadavek na přesnost se nám alespoň do určité míry podařilo (silně klesající KSPC v průběhu testů, stejně jako klesající počet neopravených chyb).

### 6.1 Budoucí práce

V rámci budoucí práce na této metodě by bylo vhodné zahrnout nějaká uživatelská nastavení (zvuk, vibrace). V ideálním případě zahrnout více layoutů písmen (protože je současný layout založen na frekvencích výskytu znaků v českém jazyce, bylo by vhodné vytvořit i layout pro anglický jazyk). Pro potřeby českého jazyka by dále bylo vhodné přidat do metody možnost psát znaky s diakritikou.

Dalším bodem do budoucna by mohlo být otestování metody na co nejvíce zařízeních se systémem Android, aby byla zjištěna míra kompatibility. Kromě toho by také bylo vhodné na multitouch klávesnici doplnit prvky, které vyžadují konvence OS Android (tlačítko, v našem případě gesto s proměnnou funkcí a podobně).

Mezi další případná menší zlepšení by se dalo zařadit vylepšení přepínání druhů klávesnice (písmena, čísla, symboly...), vylepšení grafického vzhledu metody a další.

## 7 Reference

- [1] Poláček Ondřej, (2011) Text Entry Systems [online]  
<[http://pda.felk.cvut.cz/system/assets/2036/original/pda\\_lecture\\_11\\_text\\_input.pdf?1324382378](http://pda.felk.cvut.cz/system/assets/2036/original/pda_lecture_11_text_input.pdf?1324382378)>
- [2] Onizczak A., MacKenzie S., (2004) A comparison of two input methods for keypads on mobile devices , In Proceedings of the third Nordic conference on Human-computer interaction, NordiCHI'04, 101- 104, ACM Press
- [3] Silfverberg M., MacKenzie S., Korhonen P., (2000) Predicting text entry speed on mobile phones [online] <<http://www.yorku.ca/mack/chi00.html>>
- [4] Pavlovych A., Stuerzlinger W., (2003) Less-Tap: A Fast and Easy-to-learn Text Input Technique for Phones, Department of Computer Science, York University
- [5] Castelluci S., MacKenzie S., (2008) Graffiti vs. Unistrokes: An Empirical Comparison, In Proceedings of the Human-computer interaction conference 2008, CHI'08, ACM Press
- [6] Existing Text Input Methods [online] <<http://www.sis.uta.fi/~pi52316/g/node6.html>>
- [7] Wobbrock J., (2006) EdgeWrite – A Versatile Design for Text Entry and Control, Human Computer Interaction Institute, Carnegie Mellon University
- [8] Wobbrock J., Myers B., Kembel J., (2003) EdgeWrite: A Stylus-Based Text Entry Method Designed for High Accuracy and Stability of Motion, Human Computer Interaction Institute, Carnegie Mellon University
- [9] Terry M., (2011) Input n' Stuff [online]  
<[http://hci.uwaterloo.ca/courses/CS489/f11/lectures/lecture\\_16\\_io.pdf](http://hci.uwaterloo.ca/courses/CS489/f11/lectures/lecture_16_io.pdf)>
- [10] EN Wikipedia, Swype [online] <<http://en.wikipedia.org/wiki/Swype>>

- [11] Veselý O., (2010) Vektorová klávesnice pro mobilní zařízení na platformě Android, ČVUT FEL
- [12] Pirkl M., (2011) Vector Keyboard for mobile devices based on Android platform, ČVUT FEL
- [13] MacKenzie S., Kober H., Smith D., Jones T., Skepner E., (2002) LetterWise: Prefix-based Disambiguation for Mobile Text Input, Eaton Ergonomics, ACM Press
- [14] Králík J., (2001) Czech Graphemes Frequencies [online] < <http://www.czech-language.cz/alphabet/alph-prehled.html> >

## 8 Příloha - obsah CD

- Text/** - tento dokument ve formátu pdf
- Src\_keyboard/** - zdrojové kódy aplikace (klávesnice)
- Src\_tests/** - zdrojové kódy aplikace (pomocná testovací)
- Proj/** - soubory projektů pro Eclipse
- APK/** - soubory s finální instalovatelnou verzí aplikací