

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Editor rozložení klávesnice pro X Window

Bc. Viktor Grešák

Diplomová práce

2019

ZADÁNÍ DIPLOMOVÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Viktor Grešák**
Osobní číslo: **I16214**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Editor rozložení klávesnice pro X Window**
Zadávací katedra: **Katedra softwarových technologií**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je vypracovat přehled nastavení rozložení klávesnice v prostředí X Window (zejména s ohledem na rozšíření XKB a nastavení rozložení pro více jazyků či variant) a dále přehled a hodnocení editorů klávesnic pro prostředí X Window se specifikováním vlastností a srovnáním. Zejména bude hodnocena ergonomie a úroveň podpory rozšíření XKB, a to včetně podpory skupin (různých současných rozložení) a možnosti nastavení typů jednotlivých kláves (chování ve vztahu k modifikátorům typu Shift či AltGr). Produkty budou srovnány také s podobným nástrojem pro Windows (Microsoft Keyboard Layout Creator).

V praktické části bude na základě vlastností vybrán a dle potřeby rozšířen existující editor (nebo vytvořen nový projekt) tak, aby podporoval následující vlastnosti:

- automatická detekce aktuálního nastavení rozložení kláves (import ze Serveru X),
- podpora různého nastavení typu klávesy (xkb_type),
- zobrazení až čtyř znaků na klávesách (dle typu klávesy) pro znak generovaný při stisku klávesy bez modifikátorů, Shift, AltGr a AltGr+Shift,
- definování klávesy pro generování vybraného znaku standardu Unicode zadaného pomocí hexadecimální sekvence, tabulky znaků anebo jména znaku,
- možnost nastavení fontu pro zobrazení znaků na virtuální klávesnici s detekcí ve fontu nepodporovaných znaků (ty pak případně zobrazovat vhodnou alternativou),
- vhodná podpora různých vzhledů (základní, s funkčními klávesami, s navigačními klávesami, s numerickou částí),
- případná možnost změny velikosti okna editoru,
- nezávislost na použitém GUI (KDE, GNOME, TDE, Xfce apod.), tj. funkční ve všech prostředích.

Ve vybraném editoru budou vytvořena alternativní česká rozložení s podporou běžně na klávesnici nedostupných typografických znaků (pomlčky, minus, české varianty uvozovek, výpustka, nezlomitelná mezebra, stupeň, krát, ligatury apod.). Pro vybrané znaky, které se již na klávesnici nevejdu, budou vytvořeny vhodné sekvence pro rozšíření X Compose. Vhodným způsobem bude modifikována knihovna serveru X.org zodpovědná za vkládání znaků pomocí sekvencí (X Compose) tak, aby umožňovala zadat libovolný znak standardu Unicode (např. pomocí sekvence kláves Compose, U, +, hexadecimální cifry), a to nezávisle na správci oken a jeho knihovnách (KDE, GNOME, TDE, Xfce apod.). Vytvořená česká rozložení a modifikace pro X Compose budou nabídnuta vývojářům pro začlenění do další verze projektu X.org.

Rozsah grafických prací:

Rozsah pracovní zprávy: cca 40–50 stran

Forma zpracování diplomové práce: tištěná

Seznam odborné literatury:

XKB – X Keyboard Extension [online]. X.org Foundation, 2013-09-15 [cit. 2018-10-05]. URL: <http://www.x.org/wiki/XKB/> KOSMULSKI, Micha: Creating custom keyboard layouts for X11 using XKB [online]. 2011 [cit. 2013-09-10]. URL: <http://hektor.umcs.lublin.pl/mikosmul/computing/articles/custom-keyboard-layouts-xkb.html> PASCAL, Ivan: X Keyboard Extension [online]. 2013 [cit. 2013-09-10]. URL: <http://pascal.tsu.ru/en/xkb/> TOMAN, Kamil – PASCAL, Ivan: The XKB Configuration Guide [online]. 2010 [cit. 2013-09-10]. URL: <http://www.x.org/releases/current/doc/xorg-docs/input/XKB-Config.html> PALMER, Doug: An Unreliable Guide to XKB Configuration [online]. 2004-10-11 [cit. 2013-09-10]. URL: <http://www.charvolant.org/doug/xkb/> XENITELLIS, Simos: Keyboard Layout Editor. GitHub [online]. 2008 [cit. 2016-09-19]. URL: <https://github.com/simos/keyboardlayouteditor> BERGER, Shai: Webkeys: A keyboard layout editor [online]. c2012 [cit. 2016-09-19]. URL: <http://webkeys.platonix.co.il/> The Microsoft Keyboard Layout Creator. Microsoft Developer Network [online]. Microsoft, c2016 [cit. 2016-09-19]. URL: <https://msdn.microsoft.com/en-us/globalization/keyboardlayouts.aspx> STONES, Richard – MATTHEW, Neil: Linux – Začínáme programovat. Praha: Computer Press, 2000. ISBN 80-7226-307-2. STONES, Richard – MATTHEW, Neil: Linux – Programujeme profesionálně. Praha: Computer Press, 2001. ISBN 80-7226-532-6

Vedoucí diplomové práce:

Mgr. Tomáš Hudec

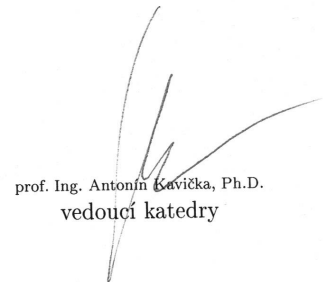
Katedra informačních technologií

Datum zadání diplomové práce: 22. října 2018

Termín odevzdání diplomové práce: 18. května 2019



Ing. Zdeněk Němec, Ph.D.
děkan



prof. Ing. Antonín Čevička, Ph.D.
vedoucí katedry

V Pardubicích dne 17. listopadu 2018

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 15. 5. 2019

Bc. Viktor Grešák

Poděkování

Děkuji vedoucímu diplomové práce Mgr. Tomáši Hudcovi za cenné rady a připomínky, které pomohly tuto práci vytvořit. Také děkuji své rodině, mé přítelkyni a přátelům za podporu.

ANOTACE

Práce se zabývá analýzou editorů rozložení klávesnice. Je vypracován přehled způsobů, kterými lze měnit rozložení klávesnice v grafickém prostředí X Window s ohledem na XKB. Hodnotí a porovnává existující editory klávesnic. Je vytvořen nový editor rozložení klávesnice pro X Window. V editoru jsou vytvořena alternativní rozložení klávesnice. Jsou navrženy sekvence X Compose umožňující zadávat znaky pomocí sekvencí kláves.

KLÍČOVÁ SLOVA

editor, klávesnice, rozložení, X, X Compose, XKB

TITLE

The Keyboard Layout Editor for X Window

ANNOTATION

The thesis analyzes keyboard layout editors. It summarizes different ways to change the keyboard layout in the X Window graphical environment taking the XKB into consideration. It compares and evaluates existing keyboard layout editors. A new keyboard layout editor for the X Window is created. Alternative keyboard layouts are created using the editor. The X Compose sequences are designed, allowing user input using multiple subsequent keystrokes.

KEYWORDS

editor, keyboard, layout, X, X Compose, XKB

OBSAH

Seznam obrázků	9
Seznam tabulek	10
Seznam zkratk	11
Úvod	12
1 Rozložení klávesnice	13
1.1 Historie	13
1.2 Typy kláves	15
1.3 Používaná rozložení klávesnice	16
2 Rozložení klávesnice v X11	21
2.1 Nástroj xmodmap	21
2.2 Nástroj localectl	23
2.3 X Keyboard Extension	24
3 Editory rozložení klávesnice	29
3.1 XKeyCaps	29
3.2 Keyboard Layout Editor	32
3.3 Microsoft Keyboard Layout Creator	35
3.4 Shrnutí	39
4 Vlastní editor klávesnice	40
4.1 Alternativní české rozložení klávesnice	47
5 Rozšíření X Compose	51
Závěr	55
Použitá literatura	57
Seznam příloh	62

A	Obsah CD	63
B	Generátor sekvencí X Compose	64

SEZNAM OBRÁZKŮ

1.1	Psací stroj Typewriter	14
1.2	Úrovně klávesy	16
1.3	Německá varianta rozložení QWERTZ	17
1.4	Rozložení klávesnice AZERTY	18
1.5	Rozložení klávesnice ÄWERTY	18
1.6	Rozložení klávesnice Dvorak s diakritikou	19
1.7	Rozložení klávesnice Colemak	19
1.8	Rozložení klávesnice Workman	20
2.1	Struktura XKB	25
2.2	Výstup příkazu <code>xkbprint</code>	27
3.1	Výběr modelu klávesnice v aplikaci XKeyCaps	29
3.2	Hlavní okno programu XKeyCaps	30
3.3	Editační okno klávesy programu XKeyCaps	31
3.4	Aplikace Keyboard Layout Editor	33
3.5	Microsoft Keyboard Layout Creator	36
3.6	Editace mrtvých kláves pomocí Microsoft Keyboard Layout Creator	37
4.1	Vzhled vlastního editoru rozložení klávesnice	41
4.2	UML diagram případů užití vytvořené aplikace	42
4.3	UML diagram aktivit popisující editaci klávesy	43
4.4	UML diagram balíčků aplikace	44
4.5	Editor s rozložením QWERTY	48
4.6	Editor s českou programátorskou variantou QWERTY	49
4.7	Editor obsahující rozložení klávesnice s typografickými znaky	50

SEZNAM TABULEK

3.1	Srovnání stávajících editorů rozložení klávesnice	39
5.1	Vytvořené sekvence pro rozšíření X Compose	52
5.2	Frameworky vstupní metody a linuxové distribuce	53
5.3	Podpora X Compose a zadávání znaků Unicode pomocí vstupní metody	54

SEZNAM ZKRATEK

ANTLR	Another Tool for Language Recognition
ASCII	American Standard Code for Information Interchange
CD	Compact Disc
ČSN	Česká technická norma
DLL	Dynamic-Link Library
Fcitx	Free Chinese Input Toy of X
GIMP	GNU Image Manipulation Program
GNOME	GNU Network Object Model Environment
GNU	GNU's Not Unix
GTK	GIMP Toolkit
GUI	Graphical User Interface
HTML	Hypertext Markup Language
IBus	Intelligent Input Bus
JDK	Java Development Kit
JSON	JavaScript Object Notation
KDE	K Desktop Environment
LED	Light-Emitting Diode
LXDE	Lightweight X11 Desktop Environment
OS	Operační systém
PDF	Portable Document Format
UIM	Universal Input Method
UML	Unified Modeling Language
XIM	X Input Method
XKB	X Keyboard Extension

ÚVOD

Moorův zákon je empirické pravidlo, které říká, že růst výpočetního výkonu je exponenciální [1]. Přestože se výpočetní technika neustále zlepšuje, hlavním prostředkem pro komunikaci s počítačem zůstává klávesnice. Může být až překvapující, jak málo se klávesnice a její nejčastěji používané rozložení – QWERTY – mění.

Dle projektu Carpalx, který si klade za cíl kvantifikovat náročnost psaní při použití rozličných rozložení klávesnic a tato rozložení optimalizovat, není QWERTY nejlepším rozložením [2]. Je možné narazit i na tvrzení, že rozložení mělo být dokonce záměrně zpomaleno (blíže viz oddíl 1.1).

Tvar a rozložení klávesnice nemá vliv pouze na rychlost psaní symbolů. Možná ještě důležitějším aspektem je pohodlí uživatele. Především častým a dlouhodobým uživatelům klávesnic (jako jsou pracovníci v kancelářích) hrozí *poškození z opakovaného namáhání*. Ať už se jedná o prevenci nebo rehabilitaci poškození, jako jsou syndrom karpálního tunelu nebo poškození šlach prstů, je účelné co nejlépe podpořit komfort uživatele.

Protože různí uživatelé rozličných profesí používají klávesnici k psaní odlišných textů, liší se i jejich nároky na kvalitní rozložení klávesnice. Po úvodní definici pojmu *rozložení klávesnice* a uvedení historie se práce zabývá existujícími rozloženými klávesnic a způsobem nastavení rozložení klávesnice v grafickém prostředí unixových systémů – X Window –, mj. jsou porovnány existující editory rozložení klávesnice.

Cílem práce je vytvoření editoru, který svými vlastnostmi umožňuje použití širokým zástupem uživatelů k vytvoření rozložení klávesnice pro X Window. V editoru jsou vytvořena alternativní rozložení, která podporují i běžně nedostupné typografické znaky. Práce se rovněž zabývá použitím X Compose, což je metoda umožňující zadávat běžně nedostupné znaky pomocí sekvencí kláves.

1 ROZLOŽENÍ KLÁVESNICE

Samotný pojem rozložení klávesnice lze chápat více způsoby. Pojem může odkazovat na fyzické uspořádání a umístění kláves. Dalším vysvětlením může být jakým způsobem klávesy vizuálně odkazují na symboly, které mají reprezentovat – tedy jaké symboly jsou umístěné na fyzických klávesách. Vizuální rozložení ovšem nemusí nutně odpovídat funkčnímu rozložení kláves, které odpovídá generovaným znakům při stisku jednotlivých kláves. Za funkční rozložení kláves může být zodpovědný operační systém, který přiřazuje číselným kódům fyzických kláves konkrétní znaky [3]. V této práci se nejčastěji používá pojem rozložení klávesnice ve smyslu funkčního rozložení kláves.

Na světě se vyskytuje množství rozložení znaků na klávesnici. Lze si položit otázky, kde se tato rozložení vzala, proč se na českých klávesnicích nejčastěji vyskytují znaky právě v uspořádání QWERTZ či jaká jsou používaná rozložení klávesnic.

1.1 Historie

Původ dnešních rozložení klávesnic lze vysledovat u psacích strojů. O rozložení QWERTY, které se dnes často používá v anglicky mluvících zemích, se v roce 1874 zasloužili pánové Christopher Latham Sholes a James Densmore [4]. V tomto roce prodali firmě E. Remington and Sons psací stroj pojmenovaný *Typewriter*.

Písmena na klávesnici psacího stroje na obrázku 1.1 jsou rozmístěna podobně, jako je tomu u dnešních klávesnic. Je možné si všimnout, že na číselném řádku chybí 1 a 0. Při psaní textu to nebyl problém, protože číslice byly nahrazovány písmeny *O* a *I* [4].

Samotné rozložení nemělo být vytvořeno s ohledem na rychlost psaní. Často některé zdroje dokonce uvádějí, že mělo být psaní na stroji záměrně zpomaleno [6, 7, 8]. Důvodem mělo být zasekávání se prvotního modelu psacího stroje při příliš rychlém psaní. Idea psacího stroje, na kterém se píše pomalu, ovšem nemusí být lákavá pro potenciální zákazníky, a tak kvůli marketingu a demonstraci rychlosti psaní na stroji měla být také všechna písmena, která tvoří slovo *Typewriter*, umístěna do stejného řádku.

Tato tvrzení však nejsou nijak podložena, a přestože jsou často zmiňována (především při snaze prosadit alternativní rozložení), skutečnost může být jiná. Pojednání z vědeckého časopisu ZINBUN z Kjótské univerzity si dalo za cíl tento údajně nepravdivý a často



Obrázek 1.1: Psací stroj Typewriter. Zdroj: [5]

omílaný příběh vyvrátit [9]. Prozkoumává počátek vzniku psacích strojů a uvádí, že mezi prvními uživateli psacího stroje byli příjemci Morseova kódu. Pokud by rozložení klávesnice psacího stroje bylo skutečně navrženo se záměrem zpomalit uživatele, nemusel by příjemce Morseova kódu stíhat zapisovat. Klávesy byly naopak umístěny tak, aby bylo možné rychle Morseův kód dekódovat a odpovídal dalším tehdejším požadavkům. Například znak I, který nahrazoval číslici 1, měl být umístěn v blízkosti čísla 8 pro rychlý zápis roku – např. „1871“. Faktem zůstává, že v roce 1888 vyhrálo toto rozložení soutěž v rychlosti psaní, což mohlo přispět k tomu, že se začalo používat jako standardní [4].

Na konci 19. století si toto rozložení klávesnice některé evropské státy upravovaly podle svých preferencí, které vycházely z konkrétních jazyků.

Rozložení momentálně používané v České republice má kořeny v Rakousku-Uhersku. V němčině se totiž Z používá mnohem častěji než Y [10]. Zatímco Z má četnost v textech přibližně 1,22 %, Y má četnost pouze cca 0,13 %. Aby zatížení levého malíčku bylo přesunuto na pravý ukazovák, došlo k záměně Y a Z. V češtině není mezi těmito znaky tak značný rozdíl. Znak Z má četnost přibližně 2,20 % a Y 1,90 % [11].

Lze soudit, že za podobou dnešního rozložení klávesnice u nás stojí spíše historický vývoj, než soustředěná optimalizace. V České republice se o problematiku rozložení znaků na klávesnici stará *Úřad pro technickou normalizaci*. Dnes je uspořádání znaků na české klávesnici popsáno normou ČSN 36 9050 [12].

1.2 Typy kláves

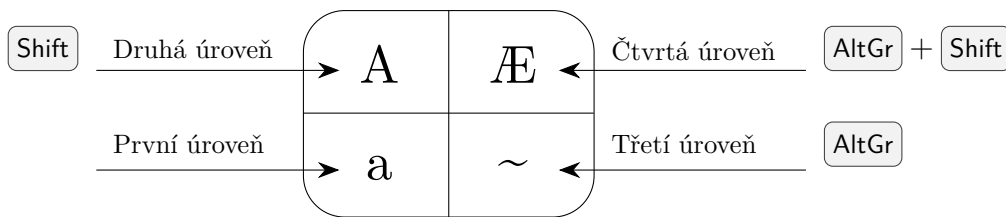
Moderní klávesnice obsahují klávesy, které lze zařadit do následujících skupin:

- znakové klávesy,
- modifikační klávesy,
- navigační klávesy,
- funkční klávesy,
- klávesy pro systémové příkazy,
- mrtvé klávesy a klávesa Compose.

Hlavní část klávesnice tvoří *znakové klávesy*, které jsou určeny pro psaní hláskových písmen, číslic, interpunkčních znamének, mezery, typografických značek a v případě cizích jazyků (např. v japonštině a korejštině) také slabičných znaků.

Modifikační klávesou se rozumí klávesa, která se používá zároveň s jinými klávesami pro změnu výchozího chování. Obvykle nejprve dojde ke stisku modifikační klávesy a poté se stiskne modifikovaná klávesa. Mezi modifikační klávesy patří například Shift (česky také přepřadovač), který se nacházel již na psacích strojích [6]. Primárně slouží k přepnutí úrovně klávesy – z minusek se stávají verzálky. Používá se ale také při klávesových zkratkách nebo například při psaní číslic u rozložení, která mají na horním řádku znaky s diakritikou.

Dalšími častými modifikačními klávesami jsou klávesy Ctrl a Alt, které se nejčastěji používají pro ovládání aplikací pomocí klávesových zkratk. Napravo od mezerníku se na českých klávesnicích také může nacházet pravý Alt – AltGr. Ten zpravidla slouží k zadávání znaků, které nejsou jinak dostupné. *Navigační klávesou* se myslí klávesa, která nějakým



Obrázek 1.2: Úrovně klávesy. Zdroj: vlastní

způsobem upravuje pozici karetu (kurzoru). Řadí se mezi ně šipky, klávesy **PgDn**, **PgUp**, **Home** nebo **End**.

V horní části klávesnice lze nalézt *funkční klávesy*. Ty jsou označeny F1 až F12 a slouží k ovládání aplikací. Význam každé klávesy závisí na operačním systému nebo konkrétním počítačovém programu. Například klávesa **F1** bývá často používána k zobrazení pomocné nabídky (nápovědy). U některých notebooků navíc bývá přítomna modifikační klávesa **Fn**, kterou lze použít v kombinaci s dalšími klávesami k ovládání zařízení (např. změna jasu displeje, změna hlasitosti reproduktorů).

Klávesy pro systémové příkazy se řadí ke klávesám ovládajícím operační systém a aplikace. Klávesu **Esc** lze používat k ukončení nebo pozastavení běhu aplikací či uzavření dialogových oken. Další klávesou z této kategorie je **Print Screen**, která slouží k pořízení snímku obrazovky. Dále sem patří například **SysRq** a **Pause/Break**. Microsoft navíc zavedl klávesy **Win** a **Menu**.

Mrtvá klávesa je blízká modifikační klávese, protože negeneruje přímo žádný znak, ale využívá se při jejich psaní. Po stisku mrtvé klávesy mění znak stisknutý po ní. Typicky se mrtvé klávesy využívají při psaní diakritických znamének – čárek, háčků, přehlásek a podobně. Klávesu **Compose** lze označit jako speciální případ mrtvé klávesy. Po jejím stisku je následující sekvence znaků (nejčastěji dva znaky) zkombinována v jediný znak. Například po postupném stisknutí kláves **Compose**, **O** a **C** je vygenerován znak ©.

1.3 Používaná rozložení klávesnice

Vyskytuje se velké množství různých rozložení klávesnic [13]. Kromě rozložení obsahujících latinku existuje řada dalších. Mnoho rozložení s latinkou pak vychází z QWERTY, přičemž

je přidána například diakritika a jsou do určité míry přemístěny nebo nahrazeny znaky tak, jak je to v daném jazyce vhodné.

QWERTZ – Rozložení široce používané ve střední Evropě. Hlavním rozdílem oproti QWERTY jsou prohozené znaky Z a Y a upravená pravá část klávesnice typicky obsahující interpunkční znaménka. Pravý alt se používá ke psaní specifických symbolů, které již není možné na klávesnici umístit (např. znaky [,], {, }, <, >, &, #, @). Toto rozložení má množství variant, které se liší podle země, ve které se používá. Kromě České republiky a Německa se toto rozložení klávesnice používá například také v Albánii, Slovinsku, Maďarsku, Polsku, Slovensku nebo Rumunsku [13].



Obrázek 1.3: Německá varianta rozložení QWERTZ. Vlastní zpracování podle [13]

AZERTY – Používaná například ve Francii a Belgii [13]. Od QWERTY se liší záměnou znaků Q s A, W se Z. Písmeno M se nachází na prostředním řádku, přičemž se na původní pozici nachází interpunkční znaménko. Podobně jako u QWERTZ lze použít pravý Alt pro psaní některých jinak nedostupných znaků. Francouzská vláda v roce 2016 začala hledat náhradu za AZERTY [14]. Rozložení bylo kritizováno zejména kvůli obtížnosti psaní a vysoké chybovosti.



Obrázek 1.4: Rozložení klávesnice AZERTY. Vlastní zpracování podle [13]

ÄWERTY – Turkmenská klávesnice má na místě Q znak Ä, Ü namísto X, Ç namísto C a Ý namísto V. Je možné si všimnout, že písmena Q, X, C a V nejsou vůbec dostupná [13].



Obrázek 1.5: Rozložení klávesnice ÄWERTY. Vlastní zpracování podle [13]

Ačkoliv QWERTY nemuselo být navrženo ke zpomalení uživatele, přece jen toto rozložení nemusí svými vlastnostmi vyhovovat požadavkům některých dnešních uživatelů. Existují proto i další alternativní rozložení. Tato rozložení připomínají QWERTY jen vzdáleně, nebo častěji vůbec. Tyto alternativy bývají často navrhovány s cílem vytvořit ergonomičtější uspořádání kláves, které redukuje pohyb rukou a umožní psát rychleji a s menším počtem chyb.

Dvorak – Mezi taková rozložení patří Dvorak. Tvůrcem je August Dvorak, americký psycholog a vzdálený příbuzný Antonína Dvořáka [7]. Je zohledněna různá síla prstů,

střídání rukou a další ergonomická pravidla na základě četností jednotlivých znaků v anglickém textu. Vzhledem k tomu, že se tato četnost v češtině oproti angličtině liší, nelze tvrdit, že je originální Dvorak nejvhodnějším možným rozložením pro psaní českých textů. Existují ovšem i české varianty obsahující navíc také diakritiku [15].



Obrázek 1.6: Rozložení klávesnice Dvorak s diakritikou. Vlastní zpracování podle [15]

Colemak – Jedná se o další oblíbené alternativní rozložení vydané v roce 2006 [16]. Tvůrce Shai Colemak se zaměřil na snadné ovládnutí uživateli QWERTY. Klávesy jsou posunuty oproti QWERTY pouze minimálně. Až na dva jsou veškeré znaky psány stejnou rukou jako na QWERTY. V souladu se studií zaměřenou na kvantifikaci úsilí psaní na klávesnici rozložení Colemak dokonce vítězí nad rozložením Dvorak [2]. Zajímavou zvláštností rozložení je chybějící klávesa `Caps Lock`, která je nahrazena klávesou `Backspace`.



Obrázek 1.7: Rozložení klávesnice Colemak. Vlastní zpracování podle [16]

Workman – Autor OJ Bucao jako programátor kvůli častému psaní začal trpět poškozením z opakovaného namáhání [17]. Při snaze ulevit svým rukám se začal poohlížet po alternativních rozložení klávesnice. Dvorak mu nevyhovoval především kvůli většímu zatížení pravého malíčku, kterého si všiml při práci s příkazy v terminálu. Poté narazil na rozložení Colemak, které ho zaujalo svými metrikami a podobností s QWERTY. Po úvodním nadšení si ale všiml problému s častým horizontálním pohybem a těžkopádným psaním v anglickém textu často se vyskytující posloupnosti písmen HE. Vytvořil proto nové rozložení, které bere v úvahu preferovaný pohyb každého prstu a snaží se o snížení pohybu zápěstí.



Obrázek 1.8: Rozložení klávesnice Workman. Vlastní zpracování podle [17]

BÉPO – Alternativní rozložení klávesnice pro francouzsky píšící uživatele [18].

Neo – Rozložení optimalizované pro němčinu [19]. Skládá se z šesti úrovní, které jsou dostupné pomocí modifikačních kláves. Kromě často používaných symbolů v němčině obsahuje v dalších úrovních znaky často používané při programování a řeckou abecedu.

ASETNIOP – Vstupní metoda mající poměrně netradiční rozložení, které využívá pouze deset kláves – osm kláves pro psaní symbolů, mezera a modifikační klávesa **Shift** [20]. Osm kláves obsahuje znaky s vysokou četností v anglickém jazyce. Pro psaní dalších symbolů včetně interpunkce a číslic jsou stisknuty kombinace kláves. Tato vstupní metoda může být vzhledem k nízkému počtu kláves zajímavá především u mobilních zařízení.

2 ROZLOŽENÍ KLÁVESNICE V X11

X Window System (také X Window, X11 nebo X) je standard pro tvorbu grafických uživatelských rozhraní [21]. Obstarává vykreslování prvků GUI a stará se o zaslání uživatelských vstupů cílovým aplikacím. Standard je založen na modelu klient-server, takže je možné pracovat s GUI aplikace, která běží na jiném počítači v síti. Je nezávislý na hardware a skládá se z více na sobě nezávislých komponent (např. X Server, Xlib). Jeho známé implementace jsou XFree86 a X.Org [22, 23].

O práci s okny se pak samostatně starají správci oken. Těch existuje více, každý mající různé vlastnosti [24]. Někteří správci se chlubí širokou paletou funkcností, jiní nízkými nároky na prostředky počítače.

Vzhledem k rozložení klávesnice zavádí X11 pojmy `keycode` a `keysym` [3]. Při stisku klávesy nebo tlačítka na myši je generován číselný kód. Tento číselný kód je následně přeložen jádrem operačního systému. Vzniká `keycode` – číselné označení reprezentující konkrétní fyzickou klávesu nebo tlačítko. Pojem `keysym` pak představuje název pro klávesu nebo tlačítko, který je zaslán aplikacím.

2.1 Nástroj `xmodmap`

Existuje více možností, pomocí kterých lze upravit rozložení klávesnice v X Window. Prvním jednoduchým nástrojem je `xmodmap` [25]. Ten umožňuje modifikovat a zobrazit současnou konfiguraci týkající se `keycode` a `keysym`. Syntaxe příkazu je následující:

```
xmodmap [-přepínače...] [soubor]
```

Soubor obsahuje seznam výrazů, které jsou příkazem zpracovány. Pro vypsání současné konfigurace lze použít příkaz ve tvaru `xmodmap -pke`. Pro uložení současné konfigurace do souboru lze použít `xmodmap -pke > nazev_souboru`. Přepínač `-pke` zajistí výpis v takovém formátu, ve kterém je možné výstup použít jako vstup příkazu (lze jej tedy využít pro uložení současného rozložení). Následuje ukázka výstupu příkazu.

```
keycode 10 = plus 1 plus 1 exclam dead_tilde exclam
keycode 11 = ecaron 2 ecaron 2 at dead_caron at
keycode 12 = scaron 3 scaron 3 numbersign dead_circumflex numbersign
```

Ukázka ilustruje přiřazení `keysym`, které reprezentují znaky tří kláves číselného řádku na české klávesnici, k číselným kódům deset až dvanáct. Manuálová stránka `xmodmap` uvádí, že

je možno každému `keycode` přiřadit až osm hodnot [25]. Tyto hodnoty odpovídají různým úrovním. Z ukázky tedy například při stisku klávesy, která odpovídá kódu 11, dojde k přiřazení `keysym ecaron`, který odpovídá znaku *ě*. Při současném stisku modifikační klávesy `Shift` je výsledný `keysym 2`. Použití pravé klávesy `Alt` pak generuje `keysym at` odpovídající znaku *@*.

Pro uvedení výrazu přímo v konzoli lze využít přepínač `-e`. Následuje příklad upravující konfiguraci tak, aby byl znak *2* psán klávesou běžně generující znak *ě* bez klávesy `Shift` a původní znak *ě* byl psán s pomocí modifikační klávesy.

```
xmodmap -e "keycode 11 = 2 ecaron"
```

Hlavičkový soubor `<X11/keysymdef.h>` obsahuje seznam `keysym`, které lze použít při práci s nástrojem `xmodmap` (v souboru jsou názvy uvedeny s prefixem `XK_`). Také je možné uvést hexadecimální kód znaku Unicode. Například následující příkaz nastaví dané klávese znak \pm :

```
xmodmap -e "keycode 10 = U00B1"
```

Kromě změny a výpisu konfigurace `keycode` a `keysym` umí nástroj také měnit konfiguraci modifikačních kláves. Nástroj `xmodmap` definuje osm modifikačních kláves s následujícími označeními: `shift`, `lock`, `control`, `mod1`, `mod2`, `mod3`, `mod4` a `mod5`. Pro výpis současné konfigurace přiřazení ke klávesám lze využít příkaz `xmodmap -pm` (viz následující ukázka).

```
xmodmap: up to 4 keys per modifier, (keycodes in parentheses):
```

```
shift    Shift_L (0x32),  Shift_R (0x3e)
lock     Caps_Lock (0x42)
control  Control_L (0x25), Control_R (0x69)
mod1     Alt_L (0x40),  Meta_L (0xcd)
mod2     Num_Lock (0x4d)
mod3
mod4     Super_L (0x85), Super_R (0x86), Super_L (0xce), Hyper_L (0xcf)
mod5     ISO_Level3_Shift (0x5c), Mode_switch (0xcb)
```

Nástroj podporuje vedle `keycode číslo = keysym...` další výrazy:

`keysym název_keysym = název_keysym...` – přiřadí všem hodnotám `keycode`, které mají aktuálně přiřazenu hodnotu `keysym` na levé straně hodnoty na pravé straně,
`add název_modifikační_klávesy = název_keysym...` – přiřadí názvy `keysym` k názvu modifikační klávesy.

`clear název_modifikační_klávesy` – odstraní veškeré položky konfigurace pro danou modifikační klávesu.

`remove název_modifikační_klávesy = název_keysym` – odstraní položky konfigurace modifikační klávesy s uvedenými `keysym`.

2.2 Nástroj `localectl`

Upravit rozložení klávesnice lze také pomocí nástroje `localectl` [26]. Tento nástroj je možné využít se `systemd` – oblíbený (pro některé však kontroverzní [27]) správce systému a služeb pro operační systémy Linux (dříve byl používán `init`). Protože je `localectl` závislý na `systemd`, nelze nástroj používat na distribucích, které využívají jinou alternativu. Příkaz lze obecně použít pro práci s místním prostředím a rozložením klávesnice. Komunikuje se systémovou službou `systemd-localed` k úpravě souborů jako jsou `/etc/locale.conf` a `/etc/vconsole.conf`.

Pro zobrazení aktuálního nastavení systémového prostředí je možné použít `localectl status` nebo pouze `localectl`. Výstup příkladu níže obsahuje nastavení místního prostředí systému ze souboru `/etc/locale.conf`, nastavení rozložení kláves ve virtuální konzoli a nastavení klávesnice pro X11.

```
System Locale: LANG=cs_CZ.UTF-8
              VC Keymap: cz-lat2
              X11 Layout: cz
              X11 Model: pc105
```

K zobrazení seznamu dostupných možných lokalizačních nastavení slouží příkaz ve tvaru `localectl list-locales`. Vzhledem k možné délce výstupu příkazu může být vhodné zkombinovat jej s příkazem `grep` – pro získání českého nastavení může příkaz vypadat takto: `localectl list-locales | grep cs`. Výstupem je nastavení ve formě použitelné pro další práci s nástrojem `localectl`, například `cs_CZ.utf8`. Samotné nastavení místního prostředí je realizováno příkazem `set-locale`. Následuje ukázka příkazu, který nastaví místní prostředí na české.

```
localectl set-locale cs_CZ.utf8
```

Podobně, jako je tomu u lokalizačního nastavení nabízí nástroj změnu rozložení klávesnice. Zobrazit dostupná rozložení je možné příkazem `localectl list-keymaps`. Příkaz

vypíše seznam možných rozložení, která mohou být uložena v adresáři `/usr/share/kbd`. Pro zobrazení českých rozložení lze využít příkazu `localectl list-keymaps | grep cz`.

Nastavení rozložení klávesnice systémové konzole a X11 pomocí nástroje `localectl` se provede příkazem `localectl set-keymap`. Následuje ukázka příkazu, který nastaví českou variantu rozložení klávesnice QWERTZ.

```
localectl set-keymap cz-qwertz
```

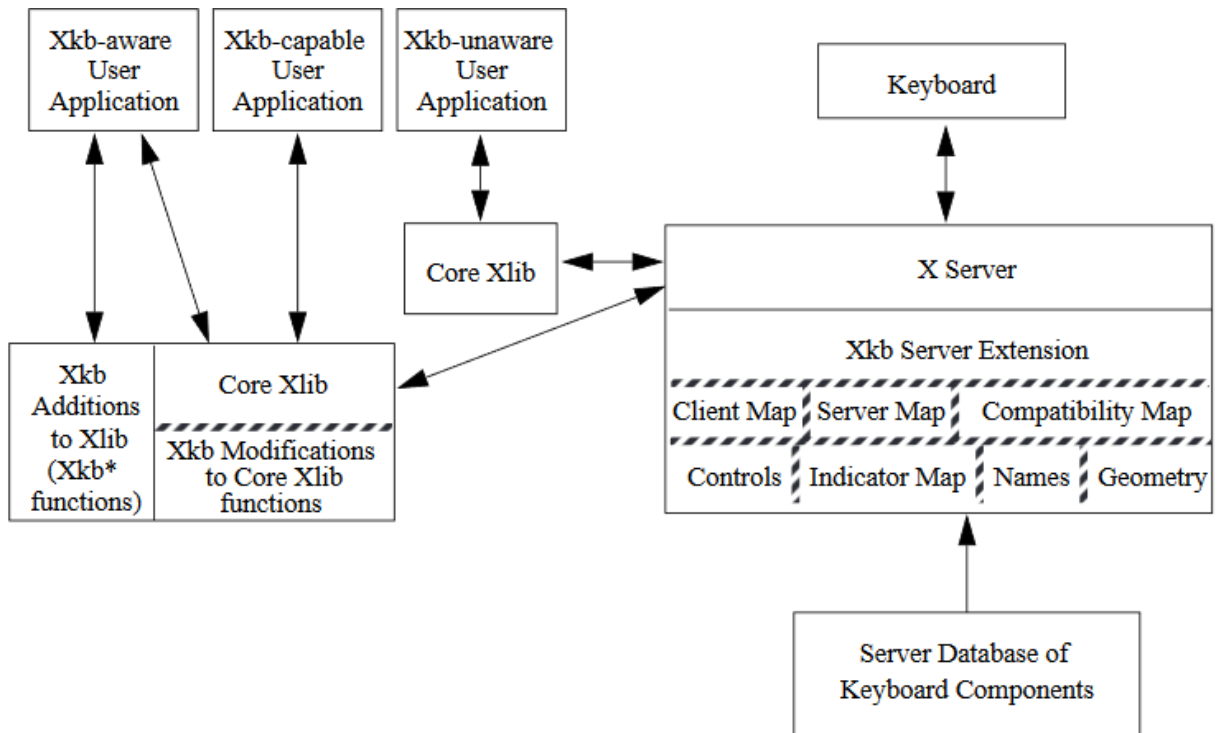
Příkaz `localectl set-x11-keymap` provádí nastavení výchozího rozložení pro X11 a virtuální konzoli. Příkaz kromě rozložení přijímá také model klávesnice, variantu rozložení a další možnosti. Pro získání použitelných argumentů lze využít příkazy:

- `list-x11-keymap-models`,
- `list-x11-keymap-layouts`,
- `list-x11-keymap-variants`
- a `list-x11-keymap-options`.

2.3 X Keyboard Extension

X Keyboard Extension, nebo také XKB, je rozšířením systému X Window, které rozšiřuje možnosti práce s klávesnicí. Je popsáno dokumentem, ze kterého se v této kapitole čerpalo [28]. Základní protokol X11 má řadu omezení, mezi které patří:

- Problémová podpora skupin klávesnice a omezení počtu skupin na dvě.
- Chabě specifikované chování přepínacích kláves jako jsou `Caps Lock` nutí implementace Serveru X k vyhledávání několika `keysym` k určení, zda se jedná o přepínací klávesu. To vede k nekompatibilnímu chování mezi Servery X.
- Limitovaná interakce mezi modifikačními klávesami. Například neexistoval spolehlivý způsob jak zjistit, zda má modifikační klávesa `Shift` zrušit aktivní přepínací klávesu.
- Chyběl jakýkoliv popis vzhledu klávesnice.




Obrázek 2.1: Struktura XKB. Zdroj: [28]

Popis klávesnice v XKB je složený z několika různých částí, se kterými lze samostatně manipulovat:

Client Map – Obsahuje konfiguraci **keycode** a **keysym**.

Server Map – Obsahuje konfiguraci, která určuje funkci kláves. Specifikuje, které klávesy jsou modifikační, jak se chovají a podobně.

Controls – Nastavení chování klávesnice, jako je opakování kláves při jejich stisku a podržení nebo modifikace pro lidi s pohybovými poruchami.

Indicators – Konfigurace chování indikátorů na klávesnici (např. LED indikátor klávesy )

Geometry – Obsahuje kompletní popis fyzického rozložení klávesnice, které lze využít pro vykreslení grafické reprezentace.

Names – Konfigurace názvů různých aspektů klávesnice jako jsou virtuální modifikační klávesy, indikátory a zvuková upozornění.

Compatibility Map – Defnuje, jak přeložit stav klávesnice základního protokolu X11 do stavu XKB. To je potřeba kvůli kompatibilitě s klienty Serveru X, které o XKB nevědí a rozšíření nepodporují.

Adresáře se soubory obsahující konfiguraci XKB jsou uloženy v adresáři X11 – například `/usr/share/X11/xkb`. Tyto soubory jsou při startu Serveru X zkompileovány pomocí nástroje `xkbcomp` do formy, kterou Server X podporující XKB umí číst [29]. Následuje popis zmíněných adresářů.

keycodes – Obsahuje soubory definující `keysym` pro `keycode` (Client Map). Část obsahu souboru je na následující ukázce.

```
<AE01> = 10;
<AE02> = 11;
<AE03> = 12;
```

types – Soubory v tomto adresáři popisují typy kláves. Typ klávesy určuje, jakým způsobem jsou ovlivňovány klávesy v závislosti na modifikačních klávesách jako jsou `Shift`. Například znakové klávesy mají přiřazen typ `ALPHABETIC`. Klávesy s tímto typem jsou ovlivňovány klávesami `Shift` a `Caps Lock`. Jiné klávesy (např. `Enter`) nereagují na modifikační klávesy, a proto mají přiřazený jiný typ – `ONE_LEVEL`.

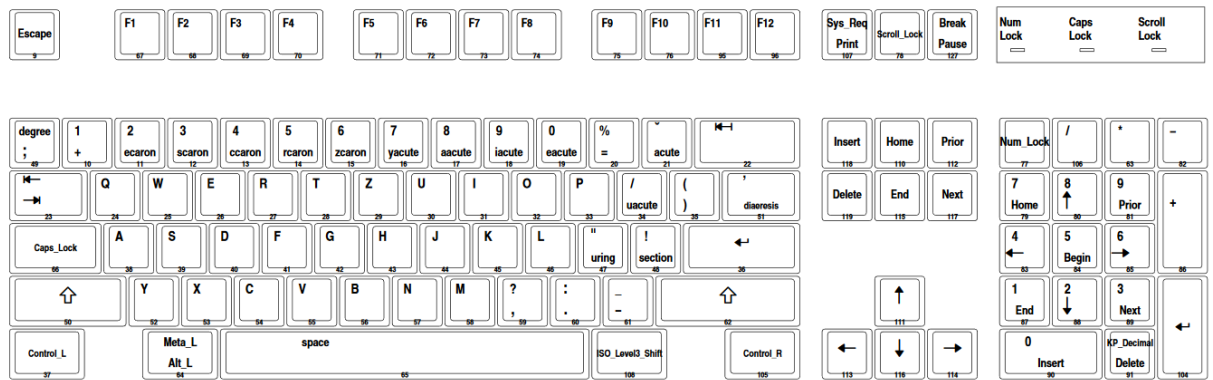
compat – Obsahuje soubory popisující chování modifikačních kláves a indikátorů LED kvůli kompatibilitě se základním protokolem X11.

symbols – Tento adresář obsahuje soubory, které přiřazují názvy definované v adresáři `keycodes` ke konkrétním hodnotám zastupující znaky. Tyto soubory jsou proto klíčové pro modifikaci a tvorbu rozložení klávesnice. Následující ukázka představuje začátek souboru pro české rozložení klávesnice. Podobně, jako je tomu u `xmodmap`, i v tomto případě lze využít hlavičkového souboru `<X11/keysymdef.h>`, který obsahuje názvy `keysym`.

```
default partial alphanumeric_keys
xkb_symbols "basic" {
    include "latin"
    name[Group1] = "Czech";
    key <TLDE> { [ semicolon, dead_abovering, grave, asciitilde ] };
    key <AE01> { [ plus, 1, exclam, dead_tilde ] };
    ...
}
```

geometry – Obsahuje soubory popisující fyzickou geometrii klávesnic. XKB tyto popisné soubory ke své činnosti nepoužívá, ale mohou být užitečné pro jiné aplikace. Následující příkaz vytvoří soubor PDF obsahující ilustraci klávesnice s českým rozložením.

```
setxkbmap cz -print | xkbcomp - - | xkbprint - - | ps2pdf - > cz.pdf
```



Obrázek 2.2: Výstup příkazu xkbprint. Zdroj: vlastní

rules – Tento adresář obsahuje množiny pravidel, které lze použít při konfiguraci rozložení klávesnice.

Konfigurace klávesnice pomocí XKB je vytvořena specifikací jednotlivých komponent, které se mají použít [30]. Specifikace spočívá ve výběru souboru pravidel a uvedením následujících parametrů:

XkbRules – Soubory s pravidly, která se mají použít pro zkombinování komponent pro konfiguraci klávesnice (např. `evdev`).

XkbModel – Model fyzické klávesnice (např. `pc105`). Modely klávesnic jsou uvedeny v souborech s pravidly.

XkbLayout – Jedno nebo více rozložení, která mají být použita (např. `cz`). Tato rozložení odpovídají souborům v adresáři `symbols`.

XkbVariant – Varianta rozložení, která má být použita. Varianty jsou definovány uvnitř souborů v adresáři `symbols`. V případě, že není varianta uvedena, je použita varianta s názvem `basic`. Varianty umožňují definici rozložení s odchylkami od základního nastavení.

XkbOptions – Další konfigurační parametry, které lze nalézt v manuálové stránce `xkeyboard-config` [31].

Definici konfigurace lze provést vytvořením nového konfiguračního souboru v adresáři `X11 - /etc/X11/xorg.conf.d/90-vlastni-rozlozeni.conf`. Následující ukázka souboru obsahuje sekci popisující vybrané rozložení, model klávesnice a navíc povoluje klávesovou kombinaci, která ukončí Server X – `Ctrl + Alt + Backspace`.

```
Section "InputClass"
    Identifier "keyboard defaults"
```

```
MatchIsKeyboard "on"  
Option "XkbLayout" "cz"  
Option "XkbModel" "pc105"  
Option "XKbOptions" "terminate:ctrl_alt_bksp"  
EndSection
```

Výše uvedené nastavení se načítá při startu Serveru X. S konfigurací klávesnice je možné také pracovat za běhu pomocí příkazu `setxkbmap`. K vypsání aktuálně používaného souboru pravidel, modelu klávesnice, rozložení klávesnice, používané varianty rozložení klávesnice a dalších nastavení slouží `setxkbmap -query`. Následuje ukázka výstupu příkazu.

```
rules:      evdev  
model:      pc101  
layout:     cz,cz  
variant:    ,qwerty  
options:    grp:alt_shift_toggle
```

Konfiguraci lze s příkazem `setxkbmap` nastavit například následovně¹:

```
setxkbmap -model pc105 -layout cz -option "terminate:ctrl_alt_bksp"
```

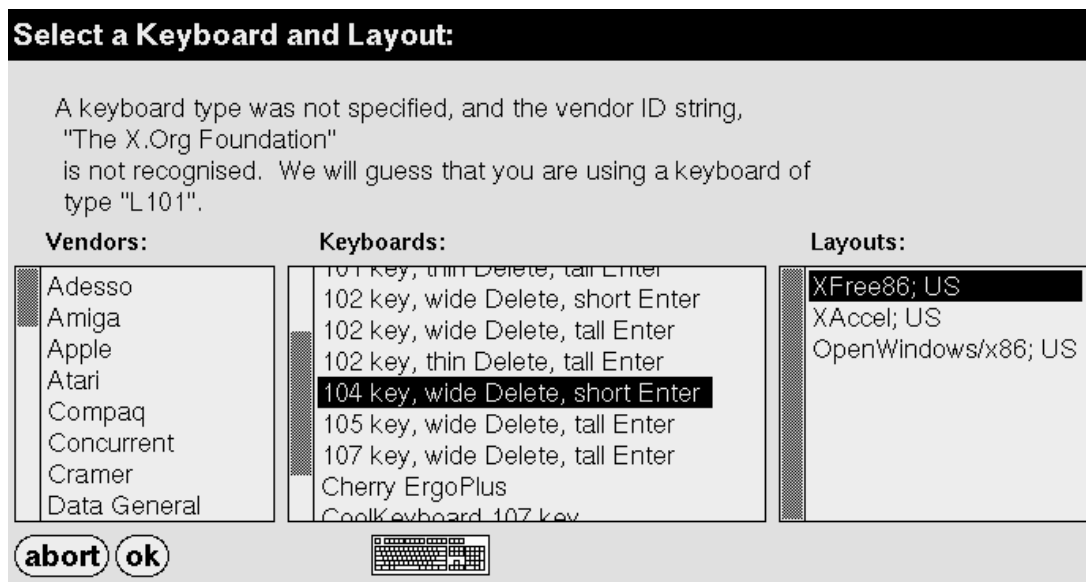
¹ Příkaz změní nastavení pouze pro uvedené přepínače. Není tedy nutné je všechny vyjmenovávat a stačí uvést např. `setxkbmap -layout cz`.

3 EDITORY ROZLOŽENÍ KLÁVESNICE

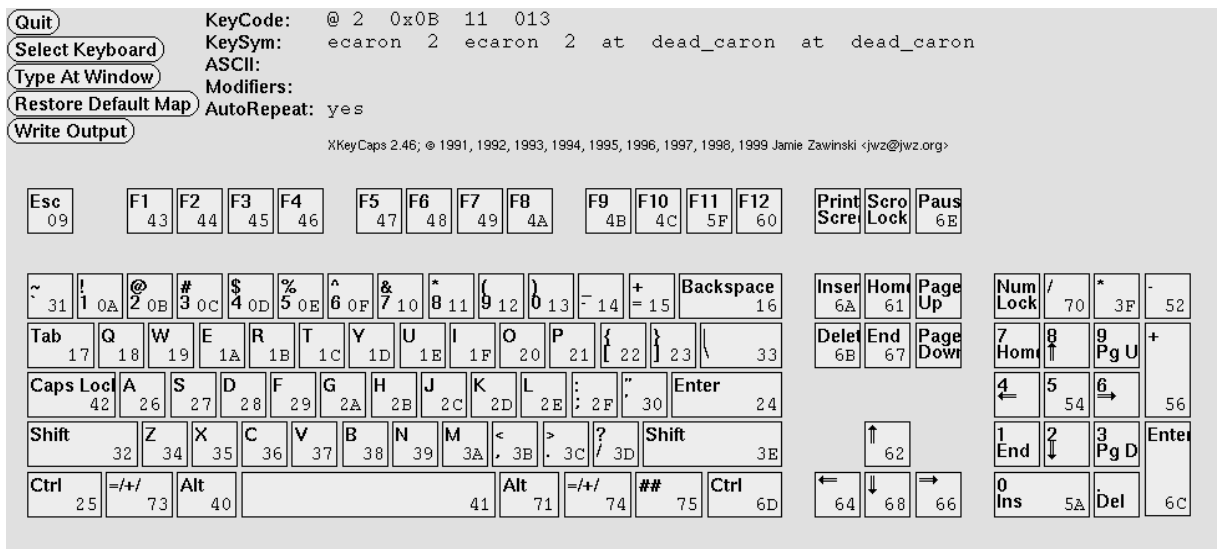
V této kapitole je vypracován přehled a hodnocení existujících editorů klávesnic pro prostředí X Window. Jsou specifikovány jejich vlastnosti a editory jsou mezi sebou porovnány. Je hodnocena ergonomie jednotlivých editorů, úroveň podpory XKB, podpora skupin (definice více rozložení, mezi kterými lze přepínat) a možnost nastavení typů jednotlivých kláves, tj. chování vzhledem k modifikačním klávesám. Tyto editory jsou rovněž srovnány s existujícím editorem pro operační systémy Microsoft Windows – Microsoft Keyboard Layout Creator.

3.1 XKeyCaps

Dnes již zastaralá aplikace představující grafickou nadstavbu nad příkazem `xmodmap` [32, 33]. Program byl vytvořen v roce 1991 a jeho vývoj byl ukončen v roce 1999. Při startu aplikace je vybrán model klávesnice – dojde k automatické detekci nebo je uživatel vyzván, aby model ručně zvolil.



Obrázek 3.1: Výběr modelu klávesnice v aplikaci XKeyCaps. Zdroj: vlastní



Obrázek 3.2: Hlavní okno programu XKeyCaps. Zdroj: vlastní

Aplikace XKeyCaps je rozdělena na tři části. V levé horní části jsou umístěna tlačítka s následujícími funkcemi:

Quit – Ukončí aplikaci.

Select Keyboard – Zobrazí dialogové okno, ve kterém uživatel může zvolit model zobrazované klávesnice. Modely klávesnic jsou rozděleny podle různých typů v levém sloupci a v pravém sloupci uživatel volí rozložení modelu. Program XKeyCaps nabízí celkem 208 různých modelů klávesnic [32].

Type At Window – Po zvolení této akce je uživatel vyzván k označení okna jiné aplikace. Poté je možné pomocí virtuální klávesnice simulovat stisknutí jednotlivých kláves – psaní do jiného okna pomocí XKeyCaps.

Restore Default Map – Obnoví výchozí nastavení klávesnice.

Write Output – Vytvoří soubor obsahující konfiguraci, kterou lze použít s příkazem `xmodmap`. Soubor je uložen do domovského adresáře a uživatel může zvolit, zda soubor bude obsahovat pouze provedené změny oproti původnímu nastavení, nebo bude v souboru obsažena kompletní konfigurace pro každou klávesu. Tento soubor lze použít ve spouštěcím skriptu, takže jsou změny nastavení klávesnice promítnuty po každém startu automaticky.

V dolní části aplikace je zobrazena virtuální klávesnice podle zvoleného modelu. Je důležité upozornit, že popisky jednotlivých kláves klávesnice jsou statické a nemusejí tedy

odpovídat reálnému aktuálnímu rozložení. V pravé dolní části každé klávesy je zobrazena hexadecimální hodnota `keycode`, kterou klávesa generuje.

Při pohybu myši přes tlačítko virtuální klávesnice nebo při stisku klávesy je zobrazen popis tlačítka v pravé horní části okna aplikace. Popis obsahuje následující informace:

KeyCode – Zobrazuje symbol zobrazený na klávese virtuální klávesnice a `keycode` generovaný při stisku klávesy v hexadecimální, desítkové a oktálové podobě.

KeySym – Seznam hodnot `keysym`, které tato klávesa generuje.

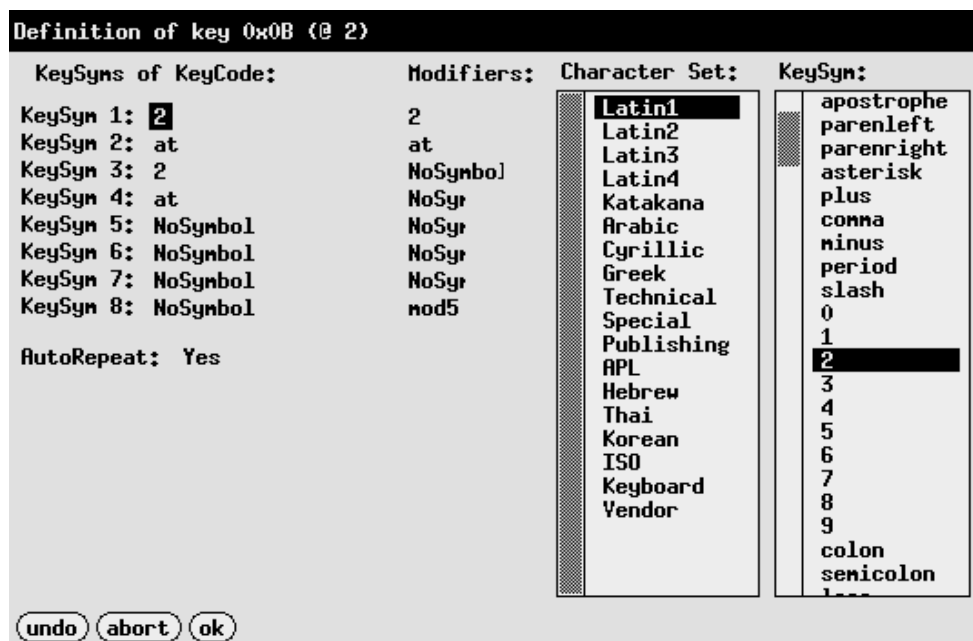
ASCII – Zobrazuje hodnotu znaku generovaného klávesou jako ASCII. Při výpisu se berou v úvahu stisknuté modifikační klávesy.

Modifiers – Zobrazí nastavený typ modifikační klávesy, např. Shift.

AutoRepeat – Poskytuje informaci o tom, zda se klávesa při držení opakuje.

S jednotlivými klávesami virtuální klávesnice lze pracovat pomocí kontextového menu, které lze vyvolat kliknutím pravého tlačítka myši. Nastavení kláves lze duplikovat, zaměňovat nebo obnovit do výchozího nastavení. Klávesu lze také vypnout, čímž se odstraní všechny hodnoty `keysym` a nastavení modifikační klávesy.

Zřejmě nejdůležitější funkcí je editace generovaných `keysym`. Tato volba zobrazí dialogové okno, ve kterém má uživatel možnost měnit až osm hodnot `keysym`. Tyto hodnoty lze vybírat ze seznamu, který je rozdělen do různých kategorií.



Obrázek 3.3: Editací okno klávesy programu XKeyCaps. Zdroj: vlastní

Hodnocení

Aplikace byla testována pod operačními systémy Arch Linux 2019.03.04 a Ubuntu 18.04. Instalace XKeyCaps je za použití správce balíčků velice snadná – `sudo pacman -Sy xkeycaps` popř. `sudo apt-get install xkeycaps`. Program používá velice zastaralou knihovnu pro vývoj aplikací GUI – X Athena Widgets [32]. Samotný program tak působí staře a nynějšího uživatele může překvapit chování GUI. Například pro vertikální posun pomocí posuvníků nelze použít standardní tažení levým tlačítkem myši. Pro pohyb dolů (o stránku) slouží levé tlačítko myši, pro pohyb nahoru pravé tlačítko myši a pro plynulý pohyb prostřední tlačítko.

Vzhledem ke stáří programu i knihovny je pochopitelné, že nemusí být zcela funkční na dnešních systémech. Ačkoliv nebyl problém spustit aplikaci na obou testovaných operačních systémech, na distribuci Arch Linux nebylo možné vybrat akci z kontextové nabídky klávesy – nebylo tedy možné upravit nastavení klávesnice. S Ubuntu tento problém nebyl. Dalším nedostatkem grafického rozhraní, který zřejmě vyplývá ze stáří softwaru, je ukončení celého programu s chybou v okamžiku, kdy uživatel uzavře dialogové okno (např. výběr modelu klávesnice) pomocí správce oken namísto tlačítek uvnitř aplikace.

Program XKeyCaps pracuje s příkazem `xmodmap` a nijak tedy nepodporuje XKB. Každé klávese lze přiřadit až osm symbolů, nicméně program nepodporuje skupiny, tj. definici více rozložení, mezi kterými lze přepínat. Přestože lze pomocí programu i na systémech používajících XKB rozložení klávesnice měnit, použití `xmodmap` se již nedoporučuje [34]. Nelze tedy ani doporučit samotný program XKeyCaps.

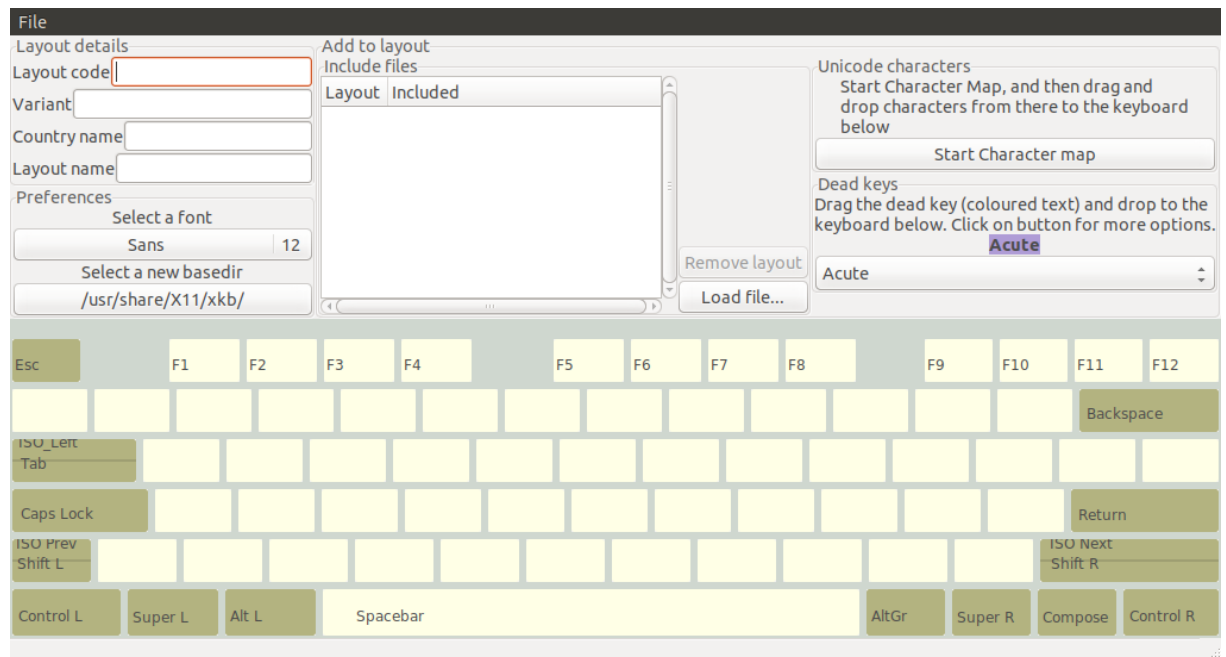
3.2 Keyboard Layout Editor

Druhým editorem rozložení klávesnice je Keyboard Layout Editor (KLE). Jedná se o aplikaci napsanou v Pythonu, která je postavená na knihovně PyGTK². Zdrojové kódy editoru lze získat na GitHubu [36, 37].

Aplikace vznikla v rámci studentského programu Google Summer of Code v roce 2008, během kterého se studenti podílejí na vývoji otevřeného softwaru [38]. Tento editor na rozdíl od předchozího pracuje s XKB a umožňuje uživateli vytvořit soubor, který

² PyGTK (dnes PyGObject) slouží ke zpřístupnění GTK v Pythonu [35]. GTK je sada nástrojů pro práci s grafickým uživatelským rozhraním.

přirazuje jednotlivým kódům kláves konkrétní hodnoty (tyto soubory lze nalézt v adresáři `/usr/share/X11/xkb/symbols`).



Obrázek 3.4: Aplikace Keyboard Layout Editor. Zdroj: vlastní

Editor rozložení klávesnice je rozdělen na dvě hlavní části. Ve vrchní části jsou umístěny ovládací prvky editoru a ve spodní části je zobrazena virtuální klávesnice. Pole editoru jsou po zapnutí prázdná. Nové rozložení klávesnice lze buď vybudovat zcela od začátku, nebo je možné načíst existující soubor symbolů pomocí položky menu **File** → **Open**. V levé horní části jsou zobrazeny podrobnosti editovaného rozložení:

Layout code – kódové označení rozložení klávesnice, např. **cz** pro české rozložení,

Variant – název editované varianty rozložení,

Country name – název země se kterou je rozložení spojeno,

Layout name – popisný název rozložení klávesnice.

Dále je možné zvolit font, který je použit pro vykreslení symbolů na virtuální klávesnici, a kořenový adresář, ve kterém se pracuje se soubory (ve výchozím nastavení ukazuje do složky s konfiguračními soubory XKB). Po pravé části jsou zobrazeny vnořené soubory. Soubory lze přidávat a lze vybírat, které varianty rozložení se mají použít.

V pravé horní části jsou umístěny prvky, které slouží k výběru symbolů. Tlačítko **Start Character Map** spustí klávesovou mapu (program **gucharmap**). Pomocí této klávesové mapy lze tažením myši umisťovat znaky na virtuální klávesnici. Pomocí rozbíjecího

seznamu níže lze vybrat mrtvou klávesu a obdobným způsobem přetáhnout na virtuální klávesnici.

Samotná virtuální klávesnice představuje alfanumerickou část klávesnice s funkčními klávesami. Klávesy zobrazují vždy čtyři úrovně přiřazených znaků. Editovat lze pouze klávesy standardně generující znaky vyjma mezerníku. Po najetí kurzorem myši na symbol a stisknutím pravého tlačítka lze vybráním volby z kontextové nabídky symbol smazat.

Hodnocení

Jako u předchozí aplikace XKeyCaps byl i tento editor testován pod operačními systémy Arch Linux 2019.03.04 a Ubuntu 18.04. Instalace editoru je poměrně složitá a vyžaduje manuální instalaci mnoha závislostí. Seznam veškerých závislostí se nepodařilo získat (zdroj [36] popisuje pouze část) a složitost instalace může být složitější v závislosti na použité distribuci Linuxu a předinstalovaných balíčcích.

Samotný program je napsán v interpretovaném jazyce Python. Zdrojové kódy ovšem nejsou kompatibilní s novou verzí Pythonu – je potřeba použít starší verzi 2. Editor také závisí na dalších balíčcích. V operačním systému Arch Linux byly balíky nainstalovány následujícím příkazem:

```
sudo pacman -Sy python2, cairo, pango, python2-gobject, python2-lxml,  
↪ python2-pip, pygtk, icu
```

V Ubuntu pak stačilo:

```
sudo apt-get install python-pip, python-gtk2, python-lxml
```

Po instalaci závislostí je nutné zpracování gramatik ANTLR³, které kód editoru používá [36]. Je možno si všimnout, že je k instalaci nutné mít nainstalované běhové prostředí Javy.

```
sudo pip install  
↪ http://www.antlr3.org/download/Python/antlr_python_runtime-3.1.2.tar.gz  
wget http://www.antlr3.org/download/antlr-3.1.2.jar  
java -classpath antlr-3.1.2.jar org.antlr.Tool *.g
```

Instalace je tedy pro běžného uživatele zbytečně složitá. Po provedení výše uvedených kroků je sice možné program spustit, stále ovšem není zcela funkční. Editor je totiž závislý na programu `gucharmap`, bez kterého nelze definovat symboly, které klávesy generují.

³ ANTLR je nástroj pro generování syntaktických analyzátorů dle definované gramatiky [39].

Grafické uživatelské prostředí působí ve srovnání s předchozím editorem moderně a kladně může být hodnocena zejména schopnost čtení a editace existujících souborů symbolů včetně práce se zanořenými soubory. Lze vytknout způsob zadávání symbolů pomocí externí znakové mapy. Při chybném přetažení symbolu uživatelem může být oprava náročná (editor neumožňuje vrátit zpět poslední editační akci).

Zásadnějším nedostatkem editoru je pevně definovaný model virtuální klávesnice a ne-možnost editace jiných než znakových kláves. Model klávesnice je popsán přímo ve zdrojovém kódu aplikace a bez jeho změny tedy není možné strukturu klávesnice měnit.

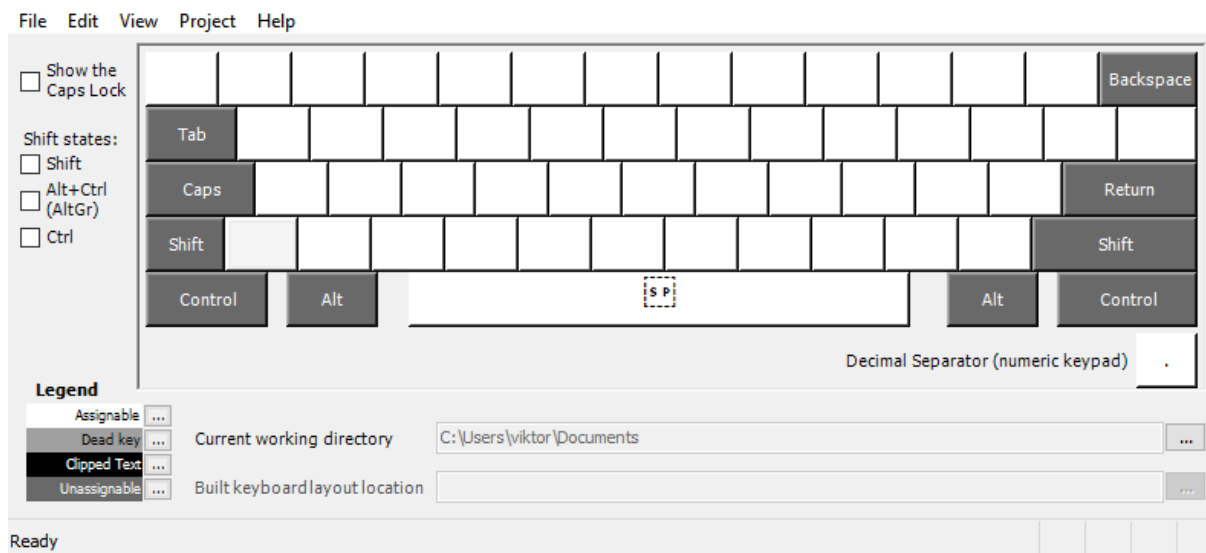
Ačkoliv je editor napsán primárně pro XKB, nepodporuje všechny jeho vlastnosti. Editor nepodporuje **skupiny** rozložení klávesnice. Nejenže nelze skupiny editovat nebo vytvářet, při načtení souboru obsahující více skupin dochází k nesprávnému chování.

Při načtení konfiguračního souboru, který obsahuje definici dvou skupin, byl vyplněn název první skupiny (**Czech**) a klávese byly přiřazeny symboly druhé skupiny – editor patrně bere v úvahu poslední definici bez ohledu na číslo skupiny. Po uložení načtené konfigurace bez jakékoliv editace jsou ostatní položky konfigurace ztraceny. Ztracena je i definice typu klávesy, kterou ani nelze v aplikaci nijak měnit. Načítaný konfigurační soubor je uveden zde.

```
xkb_symbols "basic"
{
    name[group2] = "Czech (QWERTY)";
    name[group1] = "Czech";
    key <AE09> {
        type = "FOUR_LEVEL",
        symbols[Group1] = [ iacute, 9, braceleft, dead_acute ],
        symbols[Group2] = [ iacute, 9, parenleft, dead_acute ]
    };
};
```

3.3 Microsoft Keyboard Layout Creator

Microsoft Keyboard Layout Creator (MSKLC) je editor rozložení klávesnice pro operační systémy Windows. Program je možné získat zdarma z webových stránek Microsoftu [40]. Aby bylo editor možné spustit, je nutné mít nainstalované běhové prostředí .NET 2.0. Oficiálně podporované operační systémy jsou Windows Server 2003, Windows Vista a Windows XP. Program se ale podařilo spustit i na Windows 10. Na operačním systému Windows 8.1 aplikaci spustit možné nebylo.



Obrázek 3.5: Microsoft Keyboard Layout Creator. Zdroj: vlastní

Po spuštění editoru je zobrazena prázdná virtuální klávesnice. Po levé straně je zobrazen panel, ve kterém je možné měnit úroveň editované klávesnice. Ve spodní části panelu je zobrazena legenda, která vysvětluje význam barev kláves virtuální klávesnice a umožňuje tyto barvy měnit. Legenda obsahuje vyznačení pro standardní alfanumerické znaky, mrtvé klávesy, oříznutý text a klávesy, které nelze upravovat (např. modifikační klávesy).

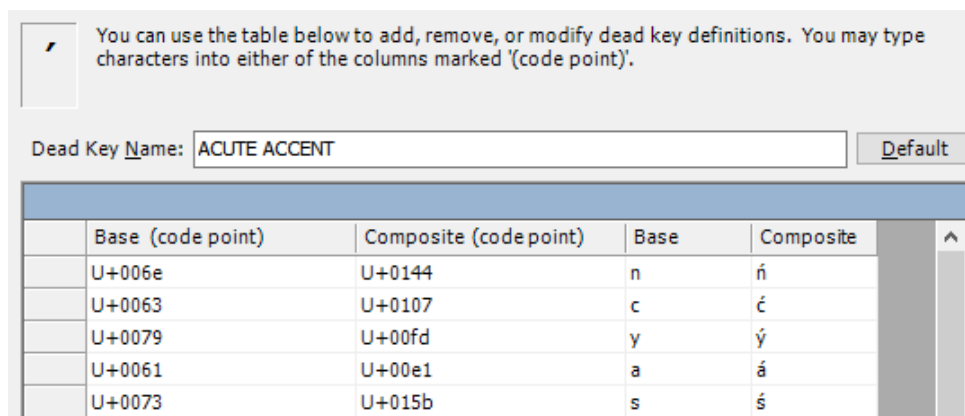
Editace rozložení klávesnice probíhá interakcí s jednotlivými klávesami virtuální klávesnice. Po kliknutí levého tlačítka myši na klávesu se zobrazí dialog umožňující editaci generovaného symbolu podle zvolené úrovně. Kliknutím pravého tlačítka myši se zobrazí kontextová nabídka s následujícími možnostmi:

Change Font... – Otevře dialog pro výběr písma pro individuální klávesu.

Properties for VK_... in all shift states – Zobrazí editační dialog umožňující změnu generovaných symbolů pro všechny úrovně (bez modifikačních kláves, s klávesou **Shift**, **AltGr** nebo kombinace **Ctrl**+**Alt** a kombinace **Shift**+**Ctrl**+**Alt**).

Properties for VK_... for '...' – Zobrazí editační dialog symbolu zvolené úrovně – stejný jako při stisku levého tlačítka myši.

Set as dead key – Označí klávesu při zvolené úrovni jako mrtvou a otevře dialog s tabulkou umožňující konfiguraci mrtvé klávesy. Tabulka definuje, které konkrétní znaky jsou generovány po stisku mrtvé klávesy a dalšího znaku.



Obrázek 3.6: Editace mrtvých kláves pomocí Microsoft Keyboard Layout Creator. Zdroj: vlastní

Dead Key Dialog... – Znovu otevře editační dialog pro konfiguraci mrtvé klávesy.

Rozložení klávesnice lze buď vytvořit od začátku celé znovu, nebo je možné načíst existující rozložení – buď zdrojový soubor generovaný editorem, nebo si uživatel může vybrat z rozložení, která jsou k dispozici v systému. Editor nabízí bohaté množství dalších funkcí jako je export aktuálního rozložení do obrázku, tisk nebo možnost nastavení fontu pro celou zobrazovanou virtuální klávesnici. Je také možné vybrat si ze tří variant modelu klávesnice, které se liší pouze umístěním a tvarem klávesy **Enter** a **Backspace** (na nastavení klávesnice nemá výběr vliv).

Aby bylo možné rozložení klávesnice navržené v editoru použít, je nejdříve nutné sestavit instalační balíček pomocí položky menu **Project** → **Build DLL and Setup Package**. Po uvedení detailů o rozložení klávesnice, jako je název, popis a cílový jazyk, jsou vygenerovány spustitelné instalační soubory do pracovního adresáře.

Hodnocení

Editor byl testován na operačním systému Windows 10 Education. Přestože není oficiálně mezi podporovanými operačními systémy, editor fungoval správně. Aplikaci se ale nepodařilo nainstalovat na Windows 8.1 Pro, takže se problémy s kompatibilitou mohou vyskytnout. Instalace editoru je snadná a pro běh aplikace nebylo vyžadováno žádné další složité nastavení systému, jako tomu bylo u předchozího editoru.

Uživatelské rozhraní je příjemné a nezatěžuje uživatele komplikovanými detaily. Přitom editor nabízí širokou paletu funkcí. Zadávání nových znaků je přímočaré a znaky lze zadávat buď sekvencí Unicode nebo přímo hodnotou znaku. Podobně jako tomu bylo

u předchozího editoru, lze i u tohoto produktu kritizovat pevně definovaný model virtuální klávesnice. Editor sice nabízí tři varianty vzhledu, nemají ovšem na funkčnost žádný dopad.

Aplikace umožňuje editaci pouze znakové části klávesnice. Nelze nastavit umístění modifikačních a dalších kláves. To může být pro některá rozložení zásadní – například rozložení Colemak a Workman používají na místě klávesy `Caps Lock` klávesu `Backspace` (viz oddíl 1.3). Rozložení klávesnice ve Windows na rozdíl od XKB nepodporuje skupiny, proto nejsou skupiny podporovány ani v editoru. Poslední výtku se týká způsobu definice mrtvých kláves. Přestože je zadávání kombinací znaků do tabulky přímočaré, může být pro uživatele zdoluhavé. Alternativou k mrtvým klávesám může být použití tzv. *kombinačních diakritických znaků* – diakritické znaky, které se píší za modifikovaný znak [41].

Pro použití vytvořeného rozložení klávesnice pro Windows 10 nestačilo spustit vygenerovaný instalační balíček. Bylo nutné upravit systémové nastavení umístěné v: **Ovládací panely** → **Hodiny, jazyk a země či oblast** → **Jazyk** → **Upřesnit nastavení**. Zde je buďto nutné pomocí rozbalovacího seznamu zvolit nainstalované rozložení v sekci **Přepsat pro výchozí metodu zadávání znaků**, nebo označit zaškrtačací pole **Použít panel jazyků na ploše, je-li k dispozici**. Toto pole pak aktivuje nabídku variant rozložení klávesnice pro zvolený jazyk ve které lze zvolit vytvořené rozložení. Nastavení nového rozložení klávesnice se při testování projevilo až po odhlášení uživatele (popř. restartu operačního systému).

3.4 Shrnutí

V předchozích kapitolách byly představeny existující editory klávesnic pro prostředí X Window a editor pro operační systémy Microsoft Windows. Pro X Window byly nalezeny pouze dva, přičemž XKeyCaps je zastaralý a nijak nepodporuje XKB – pracuje s dnes nedoporučovaným nástrojem `xmodmap`. Druhý editor s XKB pracuje, nicméně do jisté míry uživatele omezuje a nepodporuje některé důležité vlastnosti XKB – především typ kláves a definice skupin.

V následující tabulce jsou shrnuty vlastnosti editorů. Nepodařilo se nalézt editor rozložení klávesnice splňující nároky stanovené v zadání. Byl proto vytvořen nový projekt (viz další kapitola) splňující všechny porovnávané vlastnosti.

Tabulka 3.1: Srovnání stávajících editorů rozložení klávesnice. Zdroj: vlastní

Vlastnost	XKeyCaps	KLE	MSKLC
Jednoduchá instalace	✓	✗	✓
Moderní uživatelské prostředí	✗	✓	✓
Doporučeno pro dnešní OS	✗	✓	✗
Podpora XKB	✗	✓	✗
Podpora skupin	✗	✗	✗
Úprava typu klávesy	✗	✗	✗
Výběr různých modelů klávesnice	✓	✗	✗
Automatická detekce rozložení klávesnice	✓	✗	✗
Zadání znaku pomocí tabulky znaků	✓	✓	✗
Zadání libovolného znaku z Unicode	✗	✗	✓
Zadání znaku znakem (symbolem)	✗	✗	✓
Nastavení fontu	✗	✓	✓

4 VLASTNÍ EDITOR KLÁVESNICE

Po představení existujících editorů rozložení klávesnice je v této kapitole představen nově vytvořený projekt. Jedná se o aplikaci postavenou na platformě JavaFX, která pracuje s rozšířením XKB. Podobně jako tomu bylo u existujícího editoru pro XKB (viz oddíl 3.2), je primárním účelem programu vytvoření konfiguračního souboru symbolů, se kterým pracuje XKB. Zdrojové kódy editoru jsou nabídnuty volně ke stažení na GitHubu [42] a nabízí řadu dalších vlastností:

- Aktuálně používané rozložení klávesnice je automaticky detekováno (importováno ze Serveru X).
- Je podporováno zobrazení až čtyř znaků (úrovní) pro každou klávesu.
- Editor umožňuje práci s až osmi různými skupinami. Ty umožňují definici více současných rozložení, mezi kterými lze přepínat.
- Lze nastavovat typ klávesy. Ten určuje jakým způsobem je klávesa ovlivňována v závislosti na modifikačních klávesách.
- Generované znaky Unicode lze zadávat pomocí hexadecimální sekvence. Případně lze použít rozsáhlou tabulku obsahující znaky Unicode, které jsou seskupeny do 278 kategorií. Je možné uvést i jméno znaku dle X11 (tj. název `keysym` obsažený v hlavičkovém souboru `<X11/keysymdef.h>`).
- Uživatel může volit font, kterým se zobrazuje virtuální klávesnice. V případě, že aplikace detekuje vybraným fontem nepodporované znaky, pokusí se o použití alternativního fontu. Tato vlastnost umožní správné zobrazení vysokého množství znaků, které může editor zobrazovat.
- Uživatel si může zvolit model virtuální klávesnice. Může si vybrat požadovanou variantu z předem definovaných modelů nebo si může nahrát libovolný vlastní konfigurační soubor popisující strukturu klávesnice⁴. Editor tak podporuje zcela neomezené množství modelů klávesnic – aplikace je tedy vhodná i pro uživatele mající netradiční klávesnici.

⁴ Konfigurační soubor je jednoduchý textový soubor ve formátu JSON, který určuje kódy jednotlivých kláves (tj. `keycode`), jakým způsobem jsou klávesy na klávesnici rozmístěny a další vlastnosti, jako jsou odsazení, šířka a výška kláves. Uživatel může při tvorbě vlastního modelu klávesnice vycházet z existujících souborů, které jsou součástí přílohy A. Umístěny jsou ve složce `src/main/resources/model`.

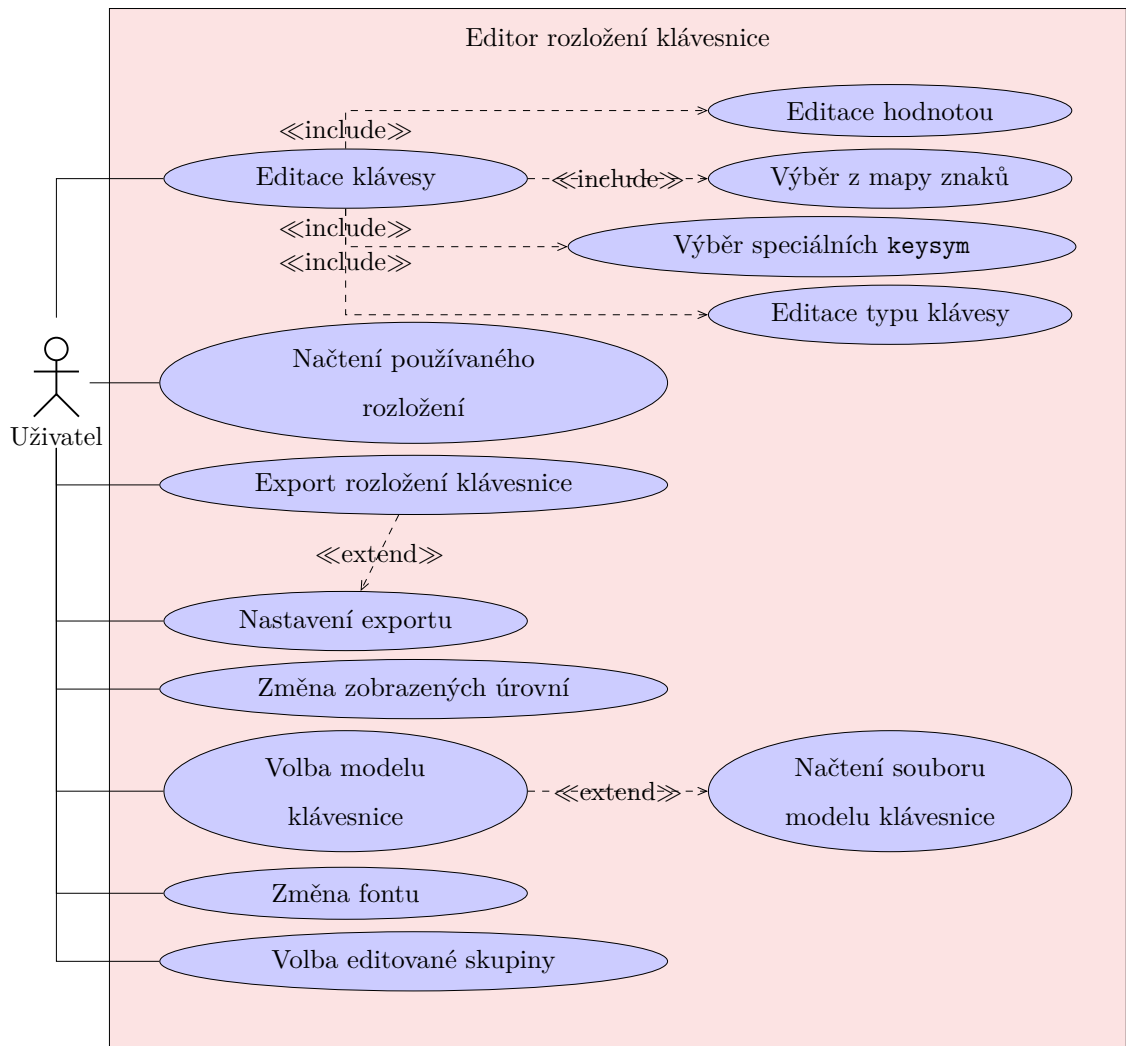
- Editor není závislý na použitém desktopovém prostředí. Nezáleží na tom, zda uživatel používá KDE, GNOME nebo jiné.
- Aplikaci lze spustit na jakékoliv platformě na které běží Java 8 a JavaFX. Ačkoliv je editor navržen primárně pro práci s XKB, je možné program spustit i na jiných platformách a vytvářet nová rozložení klávesnice.

Vzhled výsledné aplikace je ilustrován na obrázku 4.1. Při vývoji bylo dbáno na přímocharý vzhled uživatelského rozhraní, které se snadno používá a nabízí množství funkcí. Hlavní částí aplikace je virtuální klávesnice, se kterou může uživatel interagovat a volit si tak editovanou klávesu. V dolní části aplikace se nacházejí prvky rozhraní, pomocí kterých lze rozložení klávesnice měnit. Horní část editoru představuje uživatelské menu nabízející export rozložení do konfiguračního souboru XKB, nastavení exportu, nastavení vzhledu editoru a další funkce.



Obrázek 4.1: Vzhled vlastního editoru rozložení klávesnice. Zdroj: vlastní

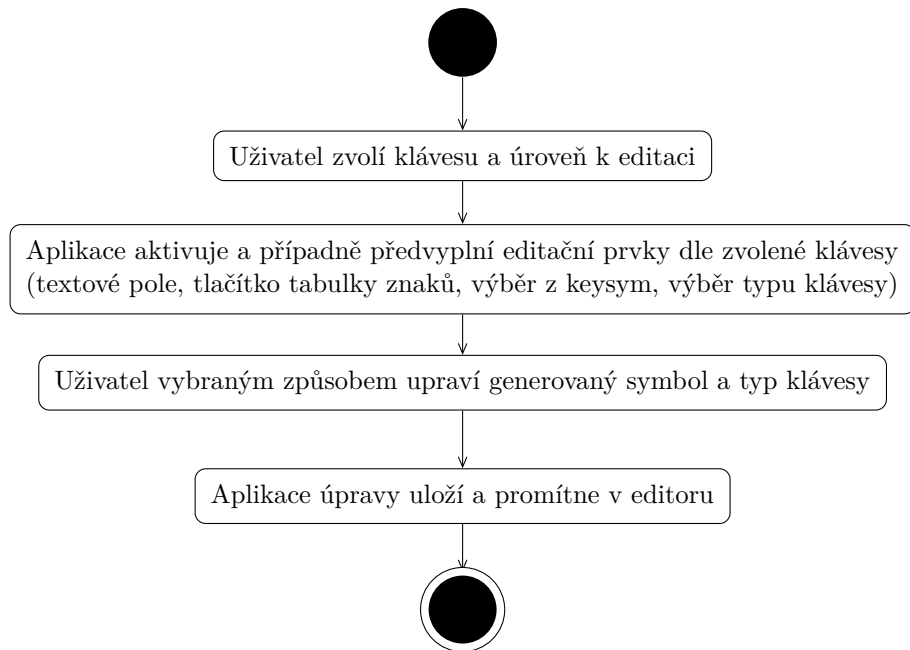
Na následujícím obrázku je zobrazen diagram případů užití. Diagram obsahuje jediného aktéra – uživatele – a případy užití, se kterými uživatel pracuje při používání vytvořeného editoru rozložení klávesnice.



Obrázek 4.2: UML diagram případů užití vytvořené aplikace. Zdroj: vlastní

Obrázek 4.3 ilustruje průběh editace klávesy. Uživatel smí zvolit zcela libovolnou klávesu na virtuální klávesnici. Není omezen editací pouze alfanumerických znaků jako tomu bylo u existujících editorů. Uživatel také volí editovanou úroveň, tj. znak generovaný při stisknutí v kombinaci s modifikačními klávesami, např. **Shift** (viz obrázek 1.2).

Uživatel má přístup k prvkům, pomocí kterých lze editovat klávesu, až po zvolení klávesy. Do editačních polí se po výběru klávesy vyplní aktuálně generovaný symbol pro zvolenou úroveň a typ klávesy. Při editaci znaků se změny projevují ve virtuální klávesnici a změny se ukládají do modelu klávesnice, který je držen v paměti a na základě kterého je později vygenerován konfigurační soubor měnící rozložení klávesnice.



Obrázek 4.3: UML diagram aktivit popisující editaci klávesy. Zdroj: vlastní

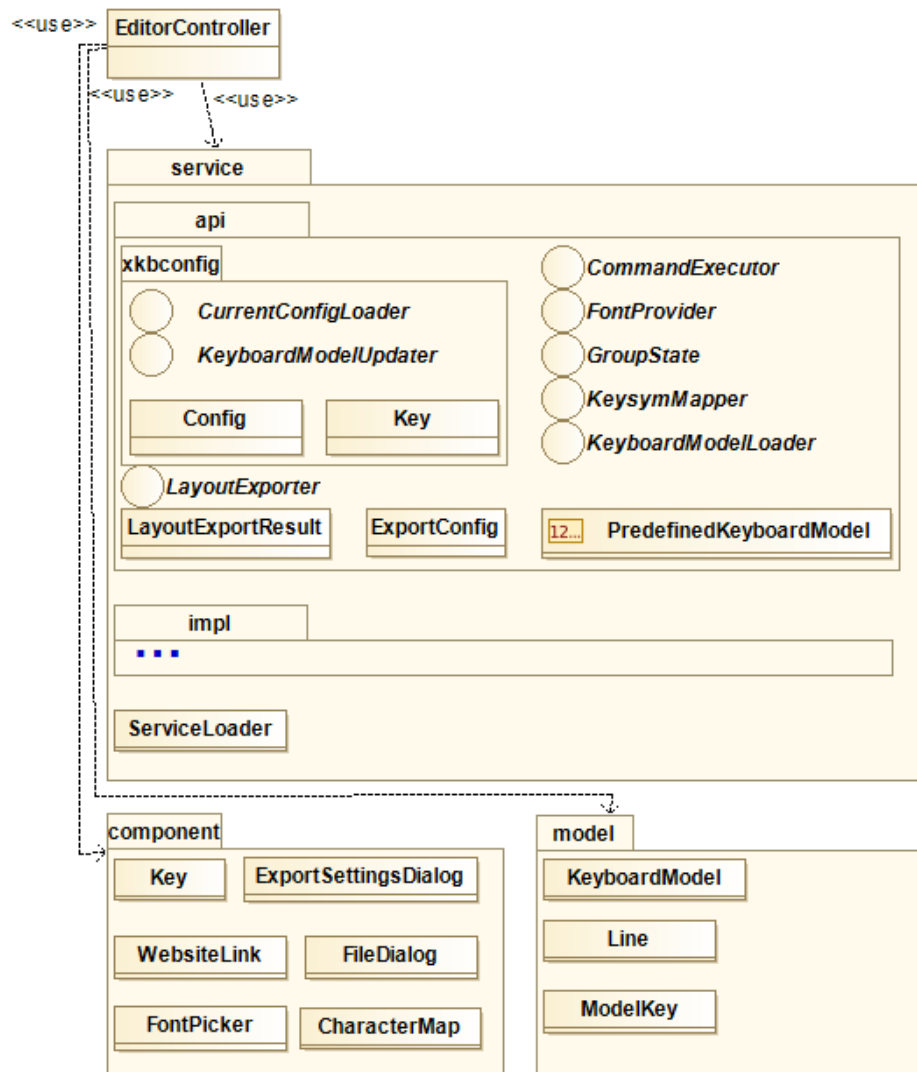
Třídy a rozhraní aplikace jsou rozděleny do tří hlavních balíčků⁵. Ústřední třídou aplikace je `EditorController`, která slouží jako řadič obsluhující požadavky uživatele a pracující se třídami, které obsahují aplikační logiku. Při startu editoru načítá model klávesnice a její aktuální rozložení pomocí volání metod patřičných tříd. Na základě načteného modelu pak dynamicky vytváří grafické uživatelské rozhraní.

Struktura aplikace je zachycena na obrázku 4.4. Kvůli stručnosti není v diagramu zachycen obsah balíčku `service.impl` – balíček obsahuje implementaci rozhraní z balíčku `service.api`. Alternativně je struktura programu zachycena v dokumentaci zdrojových kódů, která je součástí přílohy A.

Balíček `service`

Nejobsáhlejším a možná nejzajímavějším balíčkem aplikace je `service`, který se skládá ze tříd obsahujících aplikační logiku. V balíčku je možné nalézt dva další balíčky (`api` a `impl`) a třídu `ServiceLoader`. Balíček `api` obsahuje rozhraní potřebná pro realizaci

⁵ Balíček (angl. „package“) je v Javě používané označení pro jmenný prostor. Vhodné rozčlenění tříd a rozhraní do balíčků pomáhá k lepší organizaci kódu.



Obrázek 4.4: UML diagram balíčků aplikace. Zdroj: vlastní

funkcí aplikace. Balíček `impl` v sobě pak uchovává konkrétní třídy implementující tato rozhraní. Následuje popis tříd a rozhraní balíčku.

ServiceLoader – Třídy využívající balíček `service` nevyužívají implementující třídy přímo. Využívají třídu `ServiceLoader` k získání instance podle rozhraní. Tím je odstraněna přímá závislost na třídách a v případě potřeby umožněna jednoduchá plošná záměna implementujících tříd. Třída realizuje návrhový vzor *Service Locator* – jeden z návrhových vzorů řídicí se návrhovým principem tzv. *obráceného řízení* (angl. „Inversion of Control“).

Ačkoliv bývá tento návrhový vzor kritizován [43] a často se před ním dává přednost technice *vkládání závislostí* (angl. „Dependency Injection“), vzhledem k jasné struktuře

aplikace a jejímu rozsahu je dostačující. Navíc je implementace návrhového vzoru triviální – stačí uchovávat reference na konkrétní objekty v datové struktuře.

CurrentConfigLoader – Implementující třída je zodpovědná za načítání aktuální konfigurace XKB (konfigurace je obsažena ve třídě `Config`). Konfigurace obsahuje názvy skupin, možné typy kláves a zejména aktuální konfiguraci kláves. Informace jsou získány následujícím příkazem:

```
xkbcomp "$DISPLAY" -
```

V případě, že příkaz selže (např. není dostupný Server X), je použita výchozí konfigurace, která je součástí aplikace.

KeyboardModelUpdater – Stará se o aktualizaci modelu klávesnice na základě předané konfigurace.

LayoutExporter – Provede zápis předaného modelu klávesnice do specifikovaného souboru na základě konfigurace, která je uvedena ve třídě `ExportConfig`. Export je proveden ve formátu souboru symbolů, který používá XKB. Výsledek operace je obsažen ve třídě `LayoutExportResult`.

CommandExecutor – Umožňuje spuštění externího procesu pomocí předaného příkazu a získání jeho výstupu. Toto rozhraní je použito např. při získávání aktuální konfigurace a získávání aktuálně používané skupiny.

FontProvider – Zpřístupňuje užitečné metody pro práci s fontem. Umožňuje na základě vstupního textu získat název fontu, který je schopný zobrazit všechny znaky textu. Spravuje aktuálně nastavený font a umožňuje reagovat na jeho změnu. Implementující třída je také schopna vypočítat správnou velikost fontu pro daný text na základě použitého fontu a definice obdélníku (klávesy), ve kterém se text nachází.

GroupState – Udržuje informaci o vybrané skupině. Podporuje až osm skupin číslovaných od jedné. Při vzniku instance implementující třídy je načtena aktuálně vybraná skupina pomocí externího programu `xkblayout-state` [44] (pokud program není k dispozici, je vybrána první skupina). Umožňuje vybranou skupinu měnit a reagovat na její změnu.

KeysymMapper – Umožňuje přiřadit hodnotě, která určuje klávesou generovaný znak, její reprezentaci, tj. symbol zobrazený na virtuální klávesnici. Hodnota může být uživatelem zadaná nebo načtena z konfigurace a může to být buď hexadecimální hodnota určující znak Unicode ve formátu `UXXXX` či `0x100XXXX` nebo `keysym`. S hodnotami v tomto formátu pracuje XKB [45].

Pokud hodnota představuje znak Unicode, je jeho symbol získán přímo pomocí kódového bodu. Pokud se nejedná o znak Unicode, tak je hodnota symbolu získána ze souboru, který je součástí aplikace. Tento soubor obsahuje velkou část existujících `keysym` a jejich symbolů. V případě, že hodnota není nalezena ani v tomto souboru, je jako symbol použita samotná předaná hodnota.

KeyboardModelLoader – Slouží k načtení modelu klávesnice ze souboru ve formátu JSON. Soubor popisuje strukturu klávesnice. Obsahuje umístění, velikost a odsazení kláves včetně jejich kódu – `keycode`. Aplikace umožňuje výběr uživatelem specifikovaného souboru nebo možnost využít předem definovaných modelů klávesnic. Předem definované modely klávesnic jsou uvedeny ve výčtu `PredefinedKeyboardModel` a zahrnují:

- 104 kláves včetně znakové části, funkčních, modifikačních, navigačních kláves a numerickou část,
- pouze znakovou část s modifikačními klávesami,
- znakovou část s modifikačními a funkčními klávesami,
- znakovou část s modifikačními, funkčními a navigačními klávesami.

Balíček component

Balíček `component` obsahuje komponenty použité pro vybudování grafického uživatelského rozhraní.

Key – Komponenta reprezentující klávesu virtuální klávesnice. Komponenta je zodpovědná za správné vykreslení až čtyř symbolů dle zvolené konfigurace – tzn. správné umístění symbolů a použití správného fontu.

ExportSettingsDialog – Dialog obsahující nastavení týkající se exportu rozložení klávesnice. Uživatel může zvolit, které skupiny budou exportovány a zda je součástí exportované konfigurace typ kláves.

CharacterMap – Představuje tabulku znaků umožňující výběr znaku z rozsáhlého množství kategorií. Kategorie byly vytvořeny dle specifikace Unicode [46]. Tabulka znaků umožňuje také měnit font, kterým se znaky v tabulce zobrazují.

FontPicker – Umožňuje výběr fontu. Nabízí volbu rodiny fontu a jeho varianty (např. tučné písmo, kurziva). Zvolený font je v dialogu předveden na ukázkovém textu, který lze uživatelem měnit.

FileDialog – Stará se o zobrazení dialogového okna pro ukládání souborů.

WebsiteLink – Komponenta představující odkaz na webovou stránku. Po kliknutí otevře odkaz v internetovém prohlížeči.

Balíček model

Posledním balíčkem je `model`, který obsahuje třídy reprezentující model klávesnice a jeho konfiguraci.

KeyboardModel – Představuje celý model klávesnice. Obsahuje jednotlivé řádky a názvy skupin.

Line – Představuje jediný řádek klávesnice. Skládá se z jednotlivých kláves.

ModelKey – Klávesa klávesnice. Specifikuje kód klávesy (`keycode`), typ klávesy pro různé skupiny a samotné hodnoty generovaných symbolů pro různé skupiny a úrovně – `keysym` nebo hexadecimální sekvence odkazující na znaky Unicode.

4.1 Alternativní české rozložení klávesnice

Nejen pro názornou ukázkou použitelnosti představeného editoru jsou vytvořena alternativní česká rozložení. Tato rozložení byla vytvořena na základě pokynů vedoucího diplomové práce a jsou nabídnuta vývojářům pro začlenění do další verze projektu X.Org. Jsou vytvořena dvě rozložení klávesnice. Prvním rozložením je česká programátorská klávesnice. Rozložení je založené na americké variantě QWERTY. Nejprve je nastaveno rozložení a poté načteno editorem (viz obrázek 4.5). Pro nastavení rozložení byl použit následující příkaz.

```
setxkbmap -layout us -variant basic
```

Vytvořené rozložení je zobrazeno na obrázku 4.6. První řada kláves byla upravena tak, aby obsahovala znaky s českou diakritikou, jako je tomu u běžného českého rozložení klávesnice. Obsahuje tedy znaky: `+`, `ě`, `š`, `č`, `ř`, `ž`, `ý`, `á`, `í`, `é`. S modifikační klávesou `Shift` se píší původní znaky první úrovně (tj. číslice od jedné do devíti a nula) a s klávesou `AltGr` původní znaky druhé úrovně, tj. znaky `!`, `@`, `#`, `$`, `%`, `^`, `ℰ`, `*`, `(`, `)`. Při současném stisknutí kláves `Shift` a `AltGr` generují klávesy prvního řádku velká písmena s diakritikou bez nutnosti použití mrtvé klávesy nebo klávesy `Caps Lock`.



Obrázek 4.5: Editor s rozložením QWERTY. Zdroj: vlastní

Upraveny byly také klávesy prvního řádku generující *spojovník*⁶ a znak *rovná se*. Klávesa *spojovník* v nové variantě generuje znak *rovná se* tak, jako je tomu u běžného českého rozložení. Podobně jako je tomu u číslic, se i u této klávesy generují původní znaky (*spojovník* a *podtržítka*) pomocí modifikačních kláves **Shift** a **AltGr**. Čtvrtá úroveň obsahuje znak typograficky korektního *minus*. Druhá klávesa původně generující znak *rovná se* nyní obsahuje mrtvé klávesy pro čárku a háček. Třetí úroveň generuje znak *plus minus* a čtvrtá úroveň znak *nerovná se*.

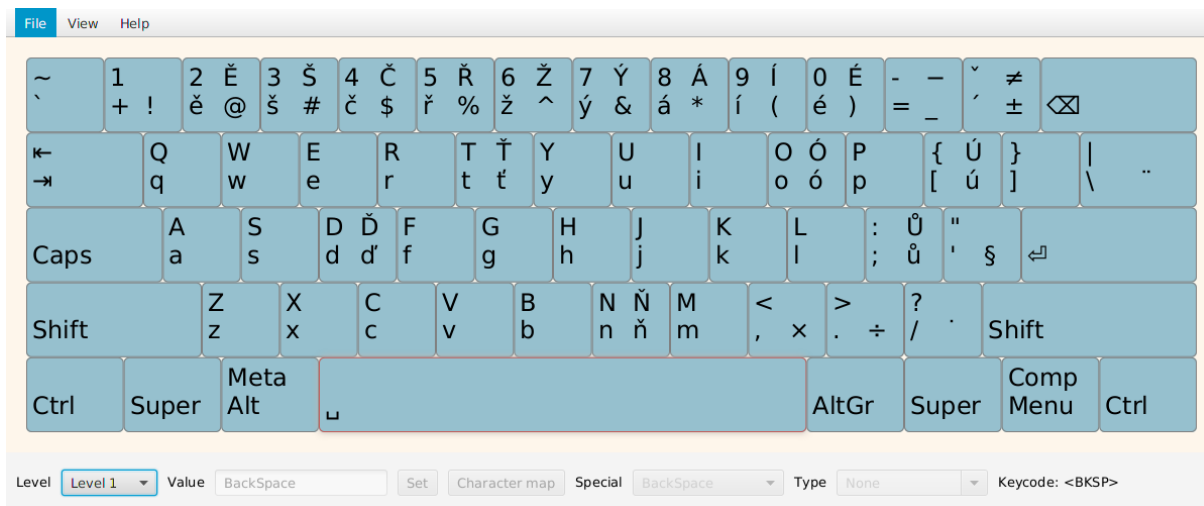
Klávesy ve znakové části klávesnice byly rozšířeny o třetí a čtvrtou úroveň o znaky s diakritikou v případě, že se na klávesnici již nevyskytují. Pravá část klávesnice byla ve třetí a čtvrté úrovni rozšířena o znaky vyskytující se na české klávesnici a znaky krát a děleno.

Rozložení klávesnice bylo pomocí editoru uloženo do souboru. Konfigurace byla přidána do souboru obsahujícího česká rozložení klávesnic – soubor nacházející se na cestě `/usr/share/X11/xkb/symbols/cz`. Ručně byl upraven název skupiny a rozložení a bylo přidáno nastavení pro klávesu **AltGr** (viz přílohu A obsahující soubor `cz`)⁷. Rozložení lze po této úpravě použít příkazem `setxkbmap -layout cz -variant prog`.

Druhé vytvořené rozložení (viz obrázek 4.7) obsahuje běžné nedostupné typografické znaky. Je vytvořena pouze část rozložení zabývající se typografickými značkami, aby

⁶ Klávesa se označuje jako *minus*, ale generuje *spojovník* (-), skutečné *minus* je stejně široké jako *plus*, je též na stejné úrovni (na matematické ose) a vypadá takto: –.

⁷ Nastavení pro klávesu **AltGr** je potřeba kvůli psaní znaků ve třetí a čtvrté úrovni. Bylo přidáno načtením standardního souboru pomocí `include "level13(ralt_switch)"`.



Obrázek 4.6: Editor s českou programátorskou variantou QWERTY. Zdroj: vlastní

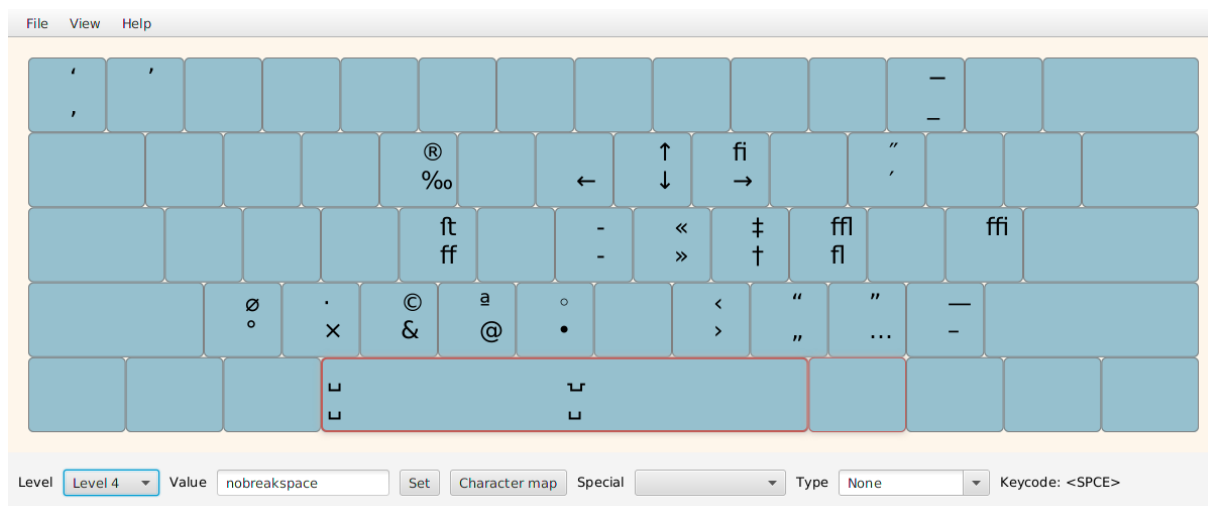
bylo možné existující rozložení klávesnice pouze rozšířit a nenutit tak uživatele používat rozložení netýkající se typografických znaků. Proto je také většina znaků definována pouze pro třetí a čtvrtou úroveň (všechny znaky vyjma mezerníku).

V levé části první řádky je umístěna jednoduchá levá dolní, horní a pravá horní *uvozovka*. Napravo je umístěn znak *podtržítka* a *minus*. Druhá řada obsahuje znak registrované ochranné známky *™*, *promile*, značky *šipek*, ligaturu *fi* a symboly pro úhlovou *minutu* a *vteřinu* (též *stopu* a *palec*).

V další řádce jsou umístěny další ligatury, obyčejný a nezlomitelný *spojovník*⁸, *francouzské uvozovky*, *kříž* a *dvojitý kříž*. V posledním řádku znakové části klávesnice jsou obsaženy znaky *stupeň*, *průměr*, *krát*, *ampersand*, značka *autorského práva*, *zavináč*, znak pro plný a prázdný *puntík* (odrážku), *jednoduché francouzské uvozovky*, *dvojitě uvozovky* (české i anglické), *výpustka*, *krátká* a *dlouhá pomlčka*. Mezerník umožňuje kromě obyčejné *mezery* tisknout také *nezlomitelnou mezeru*.

Vytvořené rozložení bylo uloženo do souboru a vygenerovaná konfigurace byla přesunuta do souboru s českými rozloženími. Jak již bylo zmíněno, rozložení s typografickými znaky je navrženo pro použití s existujícím rozložením. Následuje příkaz, který spojí obě vytvořená rozložení `cz(prog)` a `cz(typo)`, čímž vzniká česká programátorská a typografická varianta rozložení klávesnice: `setxkbmap -layout "cz(prog)+cz(typo)"`.

⁸ Jedná se o spojovníky s kódy Unicode U2010 a U2011, nikoliv základní *spojovník-minus* s kódem U002D.



Obrázek 4.7: Editor obsahující rozložení klávesnice s typografickými znaky. Zdroj: vlastní

5 ROZŠÍŘENÍ X COMPOSE

Knihovna Xlib (součást X Window) umožňuje zadávat jinak nedostupné znaky pomocí posloupnosti stisknutých `keysym`⁹ [47]. Sekvence znaků se často uvádějí speciální klávesou `Compose`. Klávesa na fyzických klávesnicích pro PC není obvyklá a proto má XKB nachystané nastavení pro použití některé stávající a zřídka používané klávesy [31]. Příkaz, který nastaví pozici klávesy `Compose` na klávesu `Menu` vypadá následovně: `setxkbmap -option compose:menu`. Jiné varianty nastavení umožňují použít klávesy `Scroll Lock`, `Pause` a další.

Uživatel tak například stiskne sekvenci kláves `Compose`, `o`, `c` a vytiskne se znak ©. Existuje množství předem definovaných sekvencí, které jsou navíc lokalizovány podle nastavení prostředí. Uživatel českého prostředí tak může mít k dispozici jiné sekvence než uživatel anglického prostředí. Konfigurace sekvencí je uložena v textových souborech `Compose` v podadresářích adresáře `/usr/share/X11/locale/`, který je dále rozčleněn podle jazyka, území a znakové sady.

Pokud chce uživatel definovat vlastní sekvence, může tak učinit vytvořením nového souboru a nastavením proměnné prostředí `$XCOMPOSEFILE`, která obsahuje cestu k souboru. Případně stačí vytvořit soubor `.XCompose` v domácím adresáři. Soubory se sekvencemi mohou načítat jiné soubory pomocí direktivy `include`. Tím je umožněno snadné navazování na již existující konfigurace. V souboru lze použít několik substitucí, pomocí kterých lze sestavit cestu:

%H – domácí adresář uživatele,

%L – výchozí lokalizovaný soubor, např. `/usr/share/X11/locale/cs_CZ.UTF-8/Compose`,

%S – bazový adresář obsahující systémové soubory `Compose`.

Jednotlivé definice sekvencí mají následující formu:

```
<keysym> [<keysym>...] : keysym|Uxxxx [# komentář]
```

Následuje krátká ukázka souboru, který využívá výchozího nastavení sekvencí a přidává sekvenci generující znak ±. Hodnota `Multi_key` je `keysym` reprezentující klávesu `Compose`. Následují další `keysym` patřící do sekvence a výsledná hodnota `plusminus`.

⁹ Při psaní sekvencí je možné použít modifikační klávesy. `Compose` pak může mít jiný význam např. pro `o`, `c` a `O`, `C`.

```
include "%L"
<Multi_key> <plus> <minus> : plusminus # ± (U00B1)
```

Vytvořené sekvence

Jako doplněk k alternativním rozložení klávesnice z předešlé kapitoly byly vytvořeny nové sekvence pro X Compose. Soubor obsahující definici sekvencí `compose_typo` je přiložen v příloze A. Protože mnoho sekvencí již existuje, jsou vytvořeny pouze dodatečné posloupnosti pro symboly, které mohou být užitečné v typografii.

Tabulka 5.1: Vytvořené sekvence pro rozšíření X Compose. Zdroj: vlastní

Sekvence kláves	Generovaný znak (keysym)
Compose, m,	Čtverčiková mezera (<code>emspace</code>)
Compose, n,	Půlčtverčiková mezera (<code>enspace</code>)
Compose, t,	Úzká mezera (<code>thinspace</code>)
Compose, h,	Vlasová mezera (<code>hairspace</code>)
Compose, n, n, b,	Úzká nezlomitelná mezera (<code>U202F</code>)
Compose, m, 3,	Mezera o velikosti třetiny čtverčiku (<code>em3space</code>)
Compose, m, 4,	Mezera o velikosti čtvrtiny čtverčiku (<code>em4space</code>)
Compose, m, 6,	Mezera o velikosti šestiny čtverčiku (<code>U2006</code>)
Compose, f,	Pevná mezera o velikosti číslice (<code>digitospace</code>)
Compose, p,	Mezera o velikosti interpunkčního znaménka (<code>punctospace</code>)
Compose, z, w,	Mezera s nulovou šířkou (<code>U200B</code>)
Compose, -, +	Znaménko \mp (<code>U2213</code>)
Compose, l, t, o, e	Symbol \leq (<code>U2266</code>)
Compose, g, t, o, e	Symbol \geq (<code>U2267</code>)
Compose, f, -	Pomlčka o velikosti číslice (<code>figdash</code>)
Compose, , <,	Krátká šipka dolů a doleva \swarrow (<code>U21B5</code>)
Compose, , , <	Dlouhá šipka dolů a doleva \searrow (<code>U21B2</code>)
Compose, o, k	Symbol zaškrtnutí (<code>U2713</code>)
Compose, b, o, k	Tučný symbol zaškrtnutí (<code>U2714</code>)
Compose, k, o	Křížek (<code>U2717</code>)
Compose, b, k, o	Tučný křížek (<code>U2718</code>)

Předcházející tabulka popisuje vytvořené sekvence a generované symboly. V první části jsou uvedeny sekvence pro rozličné typy mezer, které se v typografii používají. Dále jsou uvedeny některé matematické operátory a další symboly.

Zadávání znaků Unicode pomocí X Compose

V této kapitole je popsán způsob pomocí kterého lze psát znaky Unicode pomocí rozšíření X Compose. Generovat znaky Unicode lze ovšem v některých distribucích Linuxu ve výchozím stavu. Aplikace, které používají GTK¹⁰, umožňují zadat znak stisknutím kombinace kláves `Ctrl+Shift+u` a následným zadáním hexadecimální hodnoty [49].

Další možností je také použití tzv. *vstupní metody*. V angličtině se lze setkat s pojmy „input method“, případně „input method editor“. Hlavním účelem vstupní metody je psaní symbolů, které se nenacházejí na klávesnici – např. při psaní čínských nebo japonských textů. Implementací existuje více – patří sem např. XIM, UIM, IBus nebo Fcitx. Jednotlivé alternativy se liší filozofií zadávání znaků, náročností na systém a mírou podpory různých jazyků [50]. Některé distribuce Linuxu mohou již nějakou variantu zahrnovat [51]. Následující tabulka popisuje výchozí vstupní metody testovaných distribucí.

Tabulka 5.2: Frameworky vstupní metody a linuxové distribuce. Zdroj: vlastní

Distribuce	Výchozí desktopové prostředí	Výchozí vstupní metoda
Ubuntu 18.04 LTS	GNOME	IBus
Kubuntu 18.04 LTS	KDE	Žádná
Lubuntu 16.04 LTS	LXDE	Fcitx
Xubuntu 18.04 LTS	Xfce	Žádná

Možnost psát znaky Unicode nenabízejí všechny vstupní metody. Navíc při použití některých alternativ způsobilo nefunkčnost X Compose během testování. V některých případech pouze nebylo možné specifikovat vlastní sekvence uvedené v souboru `.XCompose`. V jiných případech nebylo možné využít ani sekvence již definované v systému. Tabulka níže shrnuje výsledky testování zmíněných vstupních metod. Vstupní metody byly konfigurovány pomocí příkazu `im-config` a testovány na linuxové distribuci Kubuntu 18.04.

¹⁰ GTK (GIMP Toolkit) je sada nástrojů pro běh grafických aplikací [48].

Tabulka 5.3: Podpora X Compose a zadávání znaků Unicode pomocí vstupní metody. Zdroj: vlastní

Vstupní metoda	Zadávání znaků Unicode	Systémové sekvence	Uživ. sekvence
XIM	✗	✓	✗
UIM	✗	✗	✗
IBus	✓	✓	✗
Fcitx	✓	✓	✓

Uživatel však nemusí používat vstupní metodu umožňující zadávání znaků Unicode. Alternativou je využití existující součásti Xlib – X Compose. Zadání této diplomové práce hovoří o rozšíření samotné knihovny Xlib. Nicméně vzhledem k plánovanému budoucímu nahrazení X Window projektem Wayland¹¹ v některých distribucích Linuxu [53] jsou po dohodě s vedoucím diplomové práce namísto toho vytvořeny sekvence, pomocí kterých lze znaky Unicode psát. Z tohoto důvodu není nabídnuto rozšíření knihovny Xlib jejím vývojářům, nicméně vytvořené sekvence jsou součástí přílohy A.

Byly vytvořeny sekvence v rozsahu hexadecimálních kódů U0000 až Uffff¹². Sekvence kláves pro zadání znaku je `Compose`, `u` a hexadecimální cifra malými písmeny. Hexadecimální cifra je vždy čtyřmístná a čísla lze zadat buď standardním numerickým řádkem nebo numerickou klávesnicí. Byl vytvořen krátký program v Javě, který vytvoří soubor `Compose` se všemi sekvencemi v uvedeném rozsahu (viz příloha B).

Hodnoty `keysym` jsou různé pro čísla zadané numerickým řádkem (0, 1, 2, ...) a numerickou klávesnicí (`KP_0`, `KP_1`, `KP_2`, ...). Proto byly vygenerovány dvě definice sekvence pro jeden znak v případě, že sekvence obsahovala číslo (netýká se kódů jako je `Uabcd`). Výsledný soubor obsahuje 129 776 různých definic. Přes vysoký počet sekvencí a poměrně vysokou velikost souboru (ca 5,8 MB) nebylo při testování vstupu zaznamenáno jakékoliv zpomalení.

¹¹ Přestože se Wayland označuje jako náhrada za X, setkává se s problémy (podpora ze strany důležitých aplikací, práce na více monitorech apod.) a například Ubuntu zatím Wayland nepoužívá jako výchozí [52].

¹² Unicode sice definuje symboly pro kódy větší než Uffff [46], ale vzhledem k tomu, že se v sekvenci X Compose nemůže objevit prefix jiné sekvence (kód U1f422 obsahuje jako prefix kód U1f42), bylo by zadávání velkého počtu hexadecimálních číslic uživatelsky nepřívětivé (např. `Compose`, `u`, `0`, `0`, `0`, `0`, `4`, `1` pro znak A).

ZÁVĚR

Cílem této diplomové práce bylo vypracování přehledu nastavení rozložení klávesnice v prostředí X Window s ohledem na rozšíření XKB, přehledu a hodnocení existujících editorů klávesnic a především tvorba nového editoru, který podporuje množství vlastností, které stávající editory nepodporují. Součástí zadání práce byla rovněž tvorba alternativních českých rozložení s podporou běžně na klávesnici nedostupných typografických znaků a vytvoření sekvencí pro rozšíření X Compose pro znaky, které se na klávesnici nevešly.

V úvodní části byl čtenář krátce seznámen s historií dnešního rozložení klávesnice. Byly uvedeny a vysvětleny v práci používané pojmy týkající se typů a úrovně kláves. Dále byly představeny některá existující rozložení klávesnic – rozložení vycházející z QWERTY i alternativní rozložení, která se snaží o ergonomičtější uspořádání kláves, jako jsou Dvorak, Colemak a jiné.

Druhá kapitola se zabývala nastavením rozložení klávesnice v prostředí X Window. Po vysvětlení pojmu X Window byly podrobně prozkoumány nástroje `xmodmap` a `localectl` umožňující měnit rozložení klávesnice. Po představení nástrojů bylo popsáno rozšíření XKB – zejména jeho výhody oproti původně používanému základnímu protokolu X11, strukturu a způsob, kterým se s XKB pracuje.

Další částí práce bylo vypracování přehledu a hodnocení existujících editorů klávesnic. Pro X Window byly nalezeny pouze dva produkty. Prvním editorem byla zastaralá aplikace XKeyCaps a druhým o něco novější Keyboard Layout Editor. Byly podrobně popsány nabízené funkce obou editorů a byla zhodnocena snadnost instalace, užívání a podpora rozšíření XKB. Bylo zjištěno, že XKeyCaps s XKB nijak nepracuje a Keyboard Layout Editor nepodporuje všechny jeho vlastnosti – např. chybí podpora skupin rozložení klávesnice nebo typu klávesy. Představen a zhodnocen byl rovněž editor rozložení klávesnice pro operační systémy Windows – Microsoft Keyboard Layout Creator. Editory byly mezi sebou porovnány a nebyl nalezen existující editor, který by splňoval kritéria stanovená v zadání diplomové práce.

Ve čtvrté kapitole byl proto prezentován nově vytvořený editor, jehož zdrojové kódy byly veřejně umístěny na GitHub. Byly uvedeny vlastnosti nového editoru, mezi které patří automatický import aktuálně používaného rozložení klávesnice, nastavitelné zobrazení úrovní kláves, podpora skupin pro definici více současných rozložení, nastavení typu klávesy

a dalších. Výčet vlastností editoru je v souladu se zadáním, a dokonce jej překonává. Vytvořená aplikace navíc nabízí možnost volby vlastního modelu virtuální klávesnice, čímž je umožněna práce i uživatelům majícím netradiční klávesnici. Pomocí diagramu byly ilustrovány případy užití aplikace a následoval popis struktury aplikace. Byly krátce popsány jednotlivé balíčky a komponenty programu.

Ve vzniklém editoru byla vytvořena dvě rozložení klávesnice splňující nároky kladené v zadání diplomové práce. První rozložení klávesnice vychází z QWERTY, přičemž obsahuje českou diakritiku. Výhodou rozložení je snadný přístup ke znakům používaných při programování a možnost psaní verzálek s diakritikou bez použití mrtvých kláves. Druhé navržené rozložení obsahuje typografické znaky. Bylo vytvořeno tak, aby jej bylo možné použít s existujícími rozloženými.

V poslední kapitole bylo představeno rozšíření X Compose umožňující zadávat znaky pomocí posloupnosti kláves. Po teoretickém úvodu a praktických ukázkách použití X Compose byly navrženy sekvence obohacující vytvořená rozložení klávesnice. Sekvence se týkaly především typografických mezer a dalších znaků podle přání vedoucího diplomové práce. Na konci se práce zabývá zadáváním znaků Unicode pomocí X Compose. Byly představeny alternativní způsoby pomocí kterých lze znaky Unicode zadávat a následně vytvořeny sekvence pro znaky v rozsahu hexadecimálních kódů U0000 až Uffff.

Ač vytvořený editor splňuje požadavky stanovené v zadání, nabízí se množství dalších možných rozšíření. Prvním je možnost načítání existujících souborů XKB. Soubory mohou obsahovat více variant rozložení a mohou vkládat jiné soubory, což by znamenalo podstatnou změnu načítání a reprezentaci dat v editoru. Dalším možným rozšířením je barevné odlišení kláves podle jejich typu. Následujícím nápadem je lokalizace programu – aplikace je momentálně pouze v angličtině. Podstatně rozsáhlejším rozšířením práce by mohla být analýza Waylandu jako plánovaného budoucího nástupce grafického prostředí X a následná úprava editoru pro práci s ním.

POUŽITÁ LITERATURA

- [1] LUNDSTROM, Mark. Moore's law forever? *Science*. 2003, **299**(5604), 210–211. ISSN 1095-9203.
- [2] KRZYWINSKI, Martin. Popular Alternatives. *Carpalx - keyboard layout optimizer* [online]. 2019? [Cit. 2019-05-10]. URL: http://mkweb.bcgsc.ca/carpalx/?popular_alternatives.
- [3] LEE, Xah. Linux: xmodmap Tutorial. *Σ Xah Code* [online]. 14. 6. 2018 [cit. 2019-05-10]. URL: http://xahlee.info/linux/linux_xmodmap_tutorial.html.
- [4] DVOŘÁK, Pavel et al. Historie klávesnice. *Česká klávesnice* [online]. 2019 [cit. 2019-05-10]. URL: <http://www.ceskaklavesnice.cz/historie>.
- [5] DADEROT. File: Remington No. 1 typewriter, made by Remington & Sons, Ilion, NY, 1873–1878, the first Sholes & Glidden model typewriter made by Remington – Wisconsin Historical Museum – DSC02806.JPG. In: *Wikimedia Commons* [online]. 29. 1. 2014 [cit. 2019-05-06]. URL: https://commons.wikimedia.org/wiki/File:Remington_No._1_typewriter,_made_by_Remington_%26_Songs,_Ilion,_NY,_1873-1878,_the_first_Sholes_%26_Glidden_model_typewriter_made_by_Remington_-_Wisconsin_Historical_Museum_-_DSC02806.JPG.
- [6] KRČMÁŘ, Petr. Historie QWERTY. *Root.cz* [online]. 24. 1. 2006 [cit. 2019-05-06]. ISSN 1212-8309. URL: <https://www.root.cz/clanky/historie-qwerty/>.
- [7] SHIPMAN, John W. Dvorak History. *Primitivism* [online]. 26. 12. 2004 [cit. 2019-05-10]. URL: <http://www.primitivism.com/dvorak.htm>.
- [8] MATRIX NETWORKS LTD. *XPeRT Keyboarding: Touch Typing Speeds without Training* [online]. 27. 9. 2009 [cit. 2019-05-10]. URL: <http://www.xpertkeyboard.com/>.
- [9] YASUOKA, Koichi a YASUOKA, Motoko. On the Prehistory of QWERTY. *ZIN-BUN* [online]. 3/2011, **42**, 161–174 [cit. 2019-05-06]. DOI: [10.14989/139379](https://doi.org/10.14989/139379).

- [10] LYONS, James. German Letter Frequencies. *Practical Cryptography* [online]. 2014 [cit. 2019-05-10]. URL: <http://practicalcryptography.com/cryptanalysis/letter-frequencies-various-languages/german-letter-frequencies/>.
- [11] KRÁLÍK, Jan. Czech Alphabet: Czech graphemes frequencies. *The Czech Language* [online]. 31. 10. 2001 [cit. 2019-05-10]. URL: <http://www.czech-language.cz/alphabet/alph-prehled.html>.
- [12] ČSN 36 9050. *Informační technika. Rozložení znaků na české klávesnici pro textové a kancelářské systémy* [online]. 1. 9. 1994 [cit. 2019-05-10]. URL: http://csnonlinefirmy.agentura-cas.cz/html_nahledy/36/16548/16548_nahled.htm. Úplné znění dostupné za poplatek.
- [13] KUČERA, Jan. *Keyboard Layout Info* [online]. 2018? [Cit. 2019-05-10]. URL: <http://kbdlayout.info/>.
- [14] DENT, Steve. France wants to fix the terrible AZERTY keyboard. *Engadget* [online]. 22. 1. 2016 [cit. 2019-05-10]. URL: <https://www.engadget.com/2016/01/22/france-wants-to-fix-the-terrible-azerty-keyboard/>.
- [15] FUČÍK, Josef. Optimální klávesnice. *ATF - psaní všemi deseti* [online]. 2019 [cit. 2019-05-10]. URL: <https://www.vsemideseti.cz/index.php?pg=17>.
- [16] COLEMAN, Shai. *Colemak keyboard layout: ergonomic, fast and easy to learn QWERTY/Dvorak alternative* [online]. 3. 2. 2019 [cit. 2019-05-10]. URL: <https://colemak.com/>.
- [17] BUCAO, OJ. *Workman Keyboard Layout: The Layout Designed with Hands in Mind* [online]. 6. 9. 2010 [cit. 2019-05-10]. URL: <https://workmanlayout.org/>.
- [18] A2 et al. Qu'est-ce que le bépo? *Disposition de clavier francophone et ergonomique bépo* [online]. 20. 4. 2018 [cit. 2019-05-10]. URL: <http://bepo.fr/wiki/Pr%C3%A9sentation>.
- [19] Neo: an Ergonomic Keyboard Layout, optimized for the German Language. *Neo – Ergonomisches Tastaturlayout* [online]. 11. 3. 2014 [cit. 2019-05-10]. URL: https://neo-layout.org/index_en.html.
- [20] POINTESA LLC. *ASETNIOP* [online]. 31. 12. 2017 [cit. 2019-05-10]. URL: <http://asetniop.com/>.

- [21] THE OPEN GROUP. The X Window System. *The Open Group* [online]. 2019? [Cit. 2019-05-10]. URL: <http://www.opengroup.org/tech/desktop/x-window-system/>.
- [22] THE XFREE86 PROJECT, Inc. *XFree86®: Home to the X Window System* [online]. 2014 [cit. 2019-05-10]. URL: <http://www.xfree86.org/>.
- [23] X.ORG FOUNDATION. *X.Org* [online]. 2. 2. 2019 [cit. 2019-05-10]. URL: <https://www.x.org/>.
- [24] CHAPMAN, Matt. *Window Managers for X* [online]. 2012? [Cit. 2019-05-10]. URL: <http://www.xwinman.org/>.
- [25] FULTON, Jim. *xmodmap: utility for modifying keymaps and pointer button mappings in X* [manuálová stránka]. 1. 6. 2014. Systém Ubuntu 18.04.
- [26] KERRISK, Michael. *localectl: Control the system locale and keyboard layout settings*. *man7.org* [online]. 11. 5. 2019 [cit. 2019-05-10]. URL: <http://man7.org/linux/man-pages/man1/localectl.1.html>.
- [27] LYNCH, Jim. Linux: Why do people hate systemd? *InfoWorld* [online]. 18. 1. 2017 [cit. 2019-05-10]. ISSN 0199-6649. URL: <https://www.infoworld.com/article/3159124/linux-why-do-people-hate-systemd.html>.
- [28] BENSON, Amber et al. The X Keyboard Extension: Library Specification. *X.Org* [online]. 10. 11. 1997 [cit. 2019-05-10]. URL: <https://www.x.org/releases/X11R7.6/doc/libX11/specs/XKB/xkblib.html>.
- [29] PASCAL, Ivan. How to configure XKB. *Ivan Pascal (Homepage)* [online]. 2018? [Cit. 2019-05-10]. URL: <http://pascal.tsu.ru/en/xkb/setup.html>.
- [30] TOMAN, Kamil a PASCAL, Ivan. The XKB Configuration Guide. *X.Org* [online]. 11/2010 [cit. 2019-05-10]. URL: <https://www.x.org/releases/X11R7.7/doc/xorg-docs/input/XKB-Config.html>.
- [31] UDALTSOV, Sergey a PASCAL, Ivan. *XKeyboard-Config: XKB data description files* [manuálová stránka]. 2018? Systém Ubuntu 18.04.
- [32] ZAWINSKI, Jamie. XKeyCaps. *juwz.org* [online]. 2. 2. 2019 [cit. 2019-05-10]. URL: <https://www.jwz.org/xkeycaps/>.

- [33] ZAWINSKI, Jamie. *XKeyCaps: graphically display and edit the X keyboard mapping* [manuálová stránka]. 6. 1. 2002. Systém Ubuntu 18.04.
- [34] GIEBEN, Miek. xmodmap is dead. *Miek Gieben* [online]. 28. 10. 2014 [cit. 2019-05-10]. URL: <https://miek.nl/2014/october/28/xmodmap-is-dead/>.
- [35] HENSTRIDGE, James a FELTMAN, Simon. PyGObject. *PyPI – the Python Package Index* [online]. 20. 3. 2019 [cit. 2019-05-10]. URL: <https://pypi.org/project/PyGObject/>.
- [36] XENITELLIS, Simos. Keyboard Layout Editor. *GitHub* [online]. 9. 10. 2018 [cit. 2019-05-10]. URL: <https://github.com/simos/keyboardlayouteditor>.
- [37] XENITELLIS, Simos. The Keyboard Layout Editor. *Mi blog lah!* [online]. 3. 10. 2008 [cit. 2019-05-10]. URL: <https://blog.simos.info/the-keyboard-layout-editor/>.
- [38] GOOGLE INTERNATIONAL LLC. *Google Summer of Code* [online]. 2019 [cit. 2019-05-10]. URL: <https://summerofcode.withgoogle.com/>.
- [39] PARR, Terence. *ANTLR* [online]. 4. 2. 2019 [cit. 2019-05-10]. URL: <https://www.antlr.org/>.
- [40] MICROSOFT CORPORATION. Microsoft Keyboard Layout Creator 1.4. *Oficiální domovská stránka Microsoft* [online]. 2019 [cit. 2019-05-10]. URL: <https://www.microsoft.com/en-us/download/details.aspx?id=22339>.
- [41] UNICODE, INC. Combining Diacritical Marks. *The Unicode Standard* [online]. 26. 4. 2019. Verze 12.1 [cit. 2019-05-10]. URL: <https://www.unicode.org/charts/PDF/U0300.pdf>.
- [42] GREŠÁK, Viktor. Keyboard layout editor. *GitHub* [online]. 7. 5. 2019 [cit. 2019-05-10]. URL: <https://github.com/vgresak/keyboard-layout-editor>.
- [43] SEEMANN, Mark. Service Locator is an Anti-Pattern. *ploeh blog* [online]. 3. 2. 2010 [cit. 2019-05-10]. URL: <https://blog.ploeh.dk/2010/02/03/ServiceLocatorisanAnti-Pattern/>.
- [44] NONPOP. xkblayout-state. *GitHub* [online]. 12. 8. 2018 [cit. 2019-05-10]. URL: <https://github.com/nonpop/xkblayout-state>.

- [45] PASCAL, Ivan. The xkb_symbols type file. *Ivan Pascal (Homepage)* [online]. 2018? [Cit. 2019-05-10]. URL: <http://pascal.tsu.ru/en/xkb/gram-symbols.html>.
- [46] UNICODE, INC. *Unicode Consortium* [online]. 2019 [cit. 2019-05-10]. URL: <https://unicode.org/>.
- [47] X.ORG FOUNDATION. *XCompose: X client mappings for multi-key input sequences* [manuálová stránka]. 2016? Systém Ubuntu 18.04.
- [48] THE GTK TEAM. *The GTK Project* [online]. 2019 [cit. 2019-05-10]. URL: <https://www.gtk.org/>.
- [49] THE GNOME PROJECT. *GtkIMContextSimple: GTK+ 3 Reference Manual. Vývojářské centrum GNOME* [online]. 11. 4. 2019 [cit. 2019-05-10]. URL: <https://developer.gnome.org/gtk3/stable/GtkIMContextSimple.html>.
- [50] WILLIS, Nathan. GNOME and input method integration. *LWN.net* [online]. 26. 6. 2012 [cit. 2019-05-10]. ISSN 1548-517X. URL: <https://lwn.net/Articles/503320/>.
- [51] THE GNOME PROJECT. *Integrated Input Methods. GNOME Library* [online]. 24. 8. 2018 [cit. 2019-05-10]. URL: <https://help.gnome.org/misc/release-notes/3.6/i18n-ibus.html.en>.
- [52] LARABEL, Michael. Canonical Reportedly Not Planning To Enable Wayland-By-Default For Ubuntu 20.04 LTS. *Phoronix* [online]. 25. 3. 2019 [cit. 2019-05-10]. URL: https://www.phoronix.com/scan.php?page=news_item&px=No-Wayland-Default-20.04-LTS.
- [53] LARABEL, Michael. Debian 10 “Buster” Switches GNOME Session To Wayland By Default. *Phoronix* [online]. 6. 9. 2017 [cit. 2019-05-10]. URL: https://phoronix.com/scan.php?page=news_item&px=Debian-Buster-GNOME-Wayland.

SEZNAM PŘÍLOH

A	Obsah CD	63
B	Generátor sekvencí X Compose	64

Příloha A: Obsah CD

Obsah CD obsahuje následující položky:

Adresář `compose` – zahrnuje soubory obsahující definice sekvencí pro rozšíření X Compose.

Soubor `compose_typo` představuje ručně vytvořené sekvence obsahující typografické znaky. Soubor `compose_unicode` obsahuje sekvence pro znaky Unicode, které byly vytvořeny pomocí krátkého programu (viz příloha B). Soubor `compose_all` obsahuje kombinaci obou.

Adresář `keyboard-layout-editor` – obsahuje veškeré zdrojové kódy vytvořeného editoru. Stejně nebo novější zdrojové kódy lze stáhnout ze stránek GitHubu.

Adresář `keyboard-layout-editor-docs` – sestává se z HTML dokumentace zdrojových kódů aplikace. Dokumentace byla vygenerovaná nástrojem `javadoc`.

Adresář `symbols` – obsahuje soubor `cz`, který v sobě uchovává definici českého rozložení klávesnice pro XKB včetně nově vytvořených variant.

Soubor `GresakV_EditorKlavesnice_TH_2019.pdf` – obsahuje text práce v elektronické podobě.

Soubor `UnicodeSequenceGenerator.java` – zdrojový kód, pomocí kterého byly vygenerovány sekvence X Compose pro znaky Unicode (viz příloha B).

Pro zkompilování zdrojových kódů, vytvoření spustitelného souboru a běh aplikace editoru je potřeba JDK 8 a JavaFX¹³. Protože byl použit Gradle – nástroj pro automatizaci sestavování programu –, je vytvoření spustitelného souboru jednoduché. Na systémech Windows stačí v příkazové řádce v adresáři se zdrojovými kódy spustit následující příkaz.

```
gradlew fatJar
```

Popřípadě na linuxových operačních systémech následující příkaz.

```
/bin/bash ./gradlew fatJar
```

Příkaz spustí úlohu, která zkompiluje kódy spolu se všemi potřebnými knihovnami do jediného souboru JAR, který se nachází v adresáři `build/libs`. Aplikace může být následně spuštěna následujícím příkazem.

```
java -jar build/libs/keyboard-layout-editor-all-1.0-SNAPSHOT.jar
```

¹³ Instrukce k instalaci OpenJDK na linuxovém operačním systému jsou včetně dalších informací součástí souboru `README.md`, který je součástí zdrojových kódů aplikace.

Příloha B: Generátor sekvencí X Compose

Krátký program obsažený ve třídě `UnicodeSequenceGenerator` vytváří po spuštění soubor `Compose` obsahující definice sekvencí znaků Unicode pro rozšíření X Compose. Program iteruje celočíselnými hodnotami v intervalu U0000 až UFFFF. Pro každou hodnotu je vytvořen řádek s definicí sekvence. Pokud hodnota v hexadecimálním tvaru obsahuje číslici od nuly do devíti, je navíc vytvořen dodatečný řádek obsahující alternativu pro numerickou klávesnici.

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.util.concurrent.atomic.AtomicBoolean;
import java.util.stream.Collectors;

public class UnicodeSequenceGenerator {
    public static void main(String[] args) throws IOException {
        try (BufferedWriter writer = new BufferedWriter(new FileWriter("Compose"))) {
            for (int codePoint = 0; codePoint <= 0xFFFF; codePoint++) {
                String codeValue = String.format("%04x", codePoint);
                String inputValues = codeValue.chars()
                    .mapToObj(ch -> String.format("<%c>", (char) ch))
                    .collect(Collectors.joining(" "));
                writer.append(String.format("<Multi_key> <u> %s : U%s%n", inputValues,
                    ↪ codeValue.toUpperCase()));
                AtomicBoolean hasKeypadAlternative = new AtomicBoolean();
                String keypadInputValues = codeValue.chars()
                    .mapToObj(ch -> {
                        if (ch <= '9') {
                            hasKeypadAlternative.set(true);
                            return String.format("<KP_%c>", (char) ch);
                        } else {
                            return String.format("<%c>", (char) ch);
                        }
                    })
                    .collect(Collectors.joining(" "));
                if (hasKeypadAlternative.get()) {
                    writer.append(String.format("<Multi_key> <u> %s : U%s%n",
                    ↪ keypadInputValues, codeValue.toUpperCase()));
                }
            }
        }
    }
}
```