

TOUCH FRIENDLY USER INTERFACE FOR SOURCE CODE EDITOR

Tereza Honzátková

Bachelor Degree Programme (4), FIT BUT

E-mail: xhonza01@stud.fit.vutbr.cz

Supervised by: Pavel Najman

E-mail: inajman@fit.vutbr.cz

Abstract: This paper describes a draft, an implementation and a testing mechanism of an alternative user interface of a C language's source code editor for touch devices. The aim of this interface is to facilitate users' work with the source code for touch devices. The resulting interface application is divided into blocks, which can be operated by the end user through drag&drop principle. The interface replaces uncomfortable software keyboard.

Keywords: UI, Android, Editor, Drag&Drop

1 ÚVOD

V současné době existuje spousta editorů zdrojových kódů na dotyková zařízení. Tyto editory se liší v množství nejrůznějších funkcí poskytovaných uživateli, grafickém rozhraní, výčtu podporovaných programovacích jazyků, ale jedno mají společné - softwarovou klávesnici zabírající mnohdy větší část obrazovky než samotný kód a psaní na nich je pomalé, nepohodlné a dochází k častým překlepům [1]. Proto můžeme najít nejrůznější alternativní klávesnice např. Swype, SwiftKey, SlideIT slibující rychlejší psaní textu.

Cílem této práce je vytvořit a zhodnotit přínos alternativního uživatelského rozhraní, pomocí kterého se bude vytvářet, editovat a ukládat zdrojový kód s minimálním použitím softwarové klávesnice. Aby bylo možné rozhraní zhodnotit a otestovat, vytvořila jsem aplikaci, pomocí které lze vytvořit jednoduchý zdrojový kód jazyka C. V práci jde primárně o uživatelské rozhraní, jsou zde tedy implementovány pouze základní prvky jazyka.

2 NÁVRH ŘEŠENÍ

Při návrhu rozhraní jsem vycházela z existujících řešení, v nichž není nutné použít klávesnici pro tvorbu kódu, jako je aplikace TouchDevelop [2] primárně vytvořená pro dotyková zařízení s operačním systémem Windows Phone a program Scratch [3]. Ačkoli je program Scratch určen pro desktopové zařízení a nebyl tedy vytvořen jako řešení problému se softwarovou klávesnicí, lze z něj také vycházet, jelikož funguje na principu snadného tvoření jednoduchých programů skládáním kusů kódu v podobě bloků do celku jejich přetahováním pomocí myši.

2.1 MYŠLENKA NAVRŽENÉHO UŽIVATELSKÉHO ROZHŘANÍ

Vzhledem k tomu, že ve zdrojových kódech používáme často stejné sekvence kódu, např. funkce, cykly, iterace, import knihoven, můžeme uživateli ulehčit práci tím, že tyto sekvence již budou v editoru nadefinované a nebude potřeba je neustále dokola psát. Dalším problémem při psaní kódu na malé klávesnici jsou speciální znaky např. ;, <, >, {, }, je tedy vhodné, aby je uživatel také nemusel složitě zadávat. Navrhla jsem tedy řešení, kde jsou různé sekvence kódu již předem definované

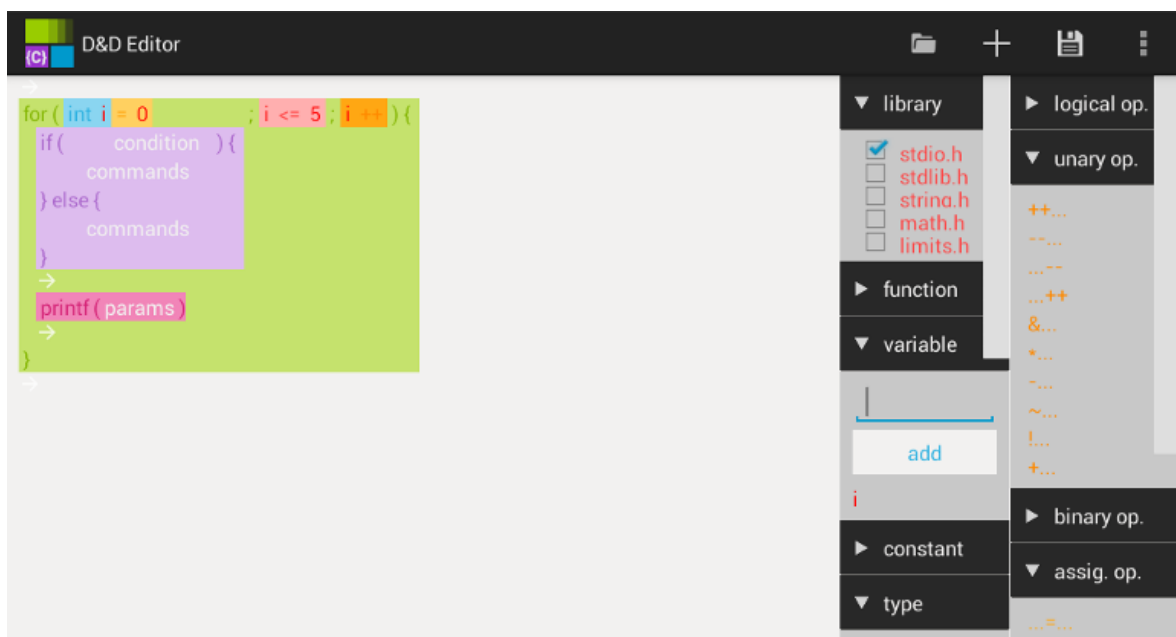
ve formě bloků a to včetně speciálních znaků. Díky tomu se nemůže stát, že zapomeneme na nějaký středník či závorku. Každý blok má také identifikátor, díky kterému je možné kontrolovat jeho umístění a hlídat tak nesmyslné kombinace, jako je například zadání cyklu do podmínky. S bloky je možné manipulovat pomocí gesta drag&drop a lze tak snadno složit v podstatě jakýkoliv kód téměř bez použití klávesnice. Tu je nutné použít pro první zadání názvů proměnných a hodnot konstant.

2.2 ROZVRŽENÍ EDITORU

Na obrázku 1 je vidět navržené uživatelské rozhraní. Jednotlivé bloky jsou zpřístupněny přes menu v logickém uspořádání do skupin:

- knihovny,
- funkce,
- proměnné,
- konstanty,
- datové typy,
- iterace,
- selekce,
- logické operátory,
- unární operátory,
- binární operátory,
- přiřazovací operátory.

Do menu je zařazen import pár základních a nejpoužívanějších knihoven postačujících pro jednoduché programy. Vzhledem k pevně dané struktuře použití knihoven není potřeba pro ně vytvářet bloky, zde zcela postačuje zaškrťávací pole. Zároveň s vybráním jakékoli knihovny se zpřístupní v položce menu s funkcemi několik vybraných funkcí z dané knihovny. Vzhledem k tomu, že aplikace slouží pouze jako nástroj pro testování použitelnosti navrženého uživatelského rozhraní, není v nynější verzi možné vkládat další knihovny s funkcemi.



Obrázek 1: Rozvržení editoru.

2.3 BLOKY A MANIPULACE GESTEM DRAG&DROP

Ovládání je založeno na principu gesta Drag&Drop a slouží pro manipulaci s bloky. Jednotlivé bloky můžeme dotykem uchopit, přetáhnout na požadované místo a zde blok upustit, tím se blok na dané místo vloží. Bloky tak můžeme přesunout z menu do prostoru a zde s nimi dále manipulovat - zanořovat, měnit jejich pořadí, odstraňovat.

Bloky lze vkládat na místa označená šipkou či nápovědou v podobě bílého textu, ty při vložení bloku zmizí. Při snaze přesunout blok na místo, kde se již jiný blok nachází, se neprovede žádná akce a oba bloky zůstávají na svém místě. Změnu pořadí bloku lze provést uchopením daného bloku a přetáhnutím na šipku v místě, kam chceme blok přemístit. Pro odstranění jakéhokoli bloku jej stačí přetáhnout zpět do menu a na pozici, z které byl blok odstraněn, se opět objeví nápověda.

2.4 TESTOVACÍ ZAŘÍZENÍ

Rozhodla jsem se uživatelské rozhraní vytvářet pro jazyk C a primárně na zařízení s parametry uvedenými v tabulce 1, ale je možné ho použít i na zařízeních s rozlišením menším. K mému rozhodnutí pro operační systém Android mne vedla data ze statistik zastoupení na trhu, pro jazyk C jsem se rozhodla kvůli jeho pevně dané struktuře.

Operační systém	Android
Verze OS	4.1.1
Typ dotykového displeje	kapacitní
Velikost displeje	10.1"
Rozlišení	1920x1200 px

Tabulka 1: Parametry testovacího zařízení.

3 ZÁVĚR

Vytvořené rozhraní bylo otestováno osmi dobrovolníky, kteří mají zkušenosti s dotykovými zařízeními. Tyto dobrovolníky jsem rozdělila do dvou skupin podle zkušenosti s programováním a nechala je přepsat stejný kód v editoru se softwarovou klávesnicí a poté s navrženým rozhraním.

Zatím jsem provedla neformální vyhodnocení, vyhodnocení formální připravuji. Z hlediska času stráveného tvorbou kódu v jednotlivých editorech není u uživatelů zkušených v programování markantní rozdíl. Jejich výsledný čas tvorby ve vytvořeném editoru se však výrazně zlepšil s každým dalším použitím. Tvorba kódu ve vytvořeném editoru však činila problém uživatelům méně zkušeným v programování, díky nízké míře znalosti jazyka C a jeho struktury, nevěděli, kde jednotlivé bloky hledat.

Ze subjektivního hodnocení zkušených uživatelů vyplývá spokojenost s tím, že v editoru s vytvořeným rozhraním uživatel nazadává středníky a závorky, tudíž na žádný nezapomene. Jako další výhodu uživatelé shledávají způsob importu knihoven, který výrazně šetří čas, a malý prostor pro případné překlipy.

Tvorba a editace zdrojového kódu v editoru s vytvořeným rozhraním ulehčí práci takovému uživateli, který věnuje určitý čas tomu, že se naučí s aplikací správně pracovat a získá určitý přehled o umístění jednotlivých bloků.

REFERENCE

- [1] MADEJ, Tomáš. Srovnání uživatelského rozhraní mobilních telefonů s dotykovým ovládním. Olomouc, 2013. Bakalářská práce. Univerzita Palackého v Olomouci, Fakulta přírodovědecká, Katedra informatiky.
- [2] Microsoft Research [online]. 1.11.2012, [cit. 2014-03-08]. Dostupné z WWW <<http://research.microsoft.com/en-us/projects/touchdevelop/>>
- [3] Resnick, M.: Scratch: Programming for all. Communications of the acm, ročník 52, č. 11, Listopad 2009: s. 60–67