

ИЗСЛЕДВАНЕ НА ДОСТЪПНОСТТА НА НЯКОИ ЧЕСТО ИЗПОЛЗВАНИ СРЕДИ ЗА РАЗРАБОТКА НА ПРОГРАМНО ОСИГУРЯВАНЕ

INVESTIGATION OF ACCESSIBILITY OF SOME FREQUENTLY USE RAPID APPLICATION DEVELOPMENT ENVIRONMENTS

Zhivko Tomov

Burgas University "A. Zlatarov"

zhmt@abv.bg

Abstract

The object of this article is to discuss some problems, that have low vision developers using some frequently use rapid application development environments. Particular attention is paid to those of them, that have free versions or are used by high schools and special secondary schools. Accessibility level of main menu, submenus, settings dialog boxes, text edit tools, diagnostic and debugging addressed interface are measured.

Made reference, that summary accessibility level of development environment DEV-C++ does not allow effective work by using screen readers and Braille displays. As better is estimated accessibility level of Microsoft Visual Studio, but, mention may here be made of some important components (related to creating GUI and database projects) are fully inaccessible. So, blind developer can without somebody else help create console applications only. Development environment C++ Builder is closely discussed, because by working with it using screen readers and Braille displays problems mentioned before does not exist. The circumstance, that this development environment is out-of-date, is not omitted, so it has not any contrivances to develop 64 bit applications and applications for Android.

The article says, that software tools for developers has low level of accessibility, then applications addressed to generally use.

Keywords: Accessibility, Blind, Low vision, Screen reader, Braille display, Developer, Programming, Microsoft Visual Studio, C++ Builder, DEV-C++

ВЪВЕДЕНИЕ

През последните две десетилетия, заедно с активизирането на формирането и прилагането на политики насочени към по-пълната интеграция на хората с увреждания в обществото, актуалност придобиха и дейностите свързани с осигуряване на достъпна обществена, битова и работна среда. В глобален аспект това са дейности насочени срещу обстоятелствата дискриминиращи хората с дефицити. Въпреки многобройните нормативни мерки и усилията по прилагането им [6], пътят към достъпността в различните ѝ аспекти далеч не е извървян. Доказателство за това са, и липсващите в много обществени сгради рампи и тактилни пътеки, и практически повсеместното отсъствие на тифло коментиране в музеите, театрите и кинозалоните, и текстът срещан в телевизионни предавания - „Потърсете ни! Телефонният номер е в долния десен ъгъл на екрана“. Бурното навлизане на дигитализацията в ежедневието на огромното

мнозинство от населението на света постави особено остро въпроса за достъпността на всичките и продукти и аспекти. Ред законодателни мерки предписват на създателите на дигитални средства да съобразяват функционалността и интерфейса на продукцията си със специфичните възможности на потребителите с дефицити. Познавайки се на ред международни стандарти, Н. Събев [2] отбелязва: „Още първото изискване е „проектирането и разработката на софтуерните компоненти и потребителските интерфейси да спазват стандартите за достъпност WCAG 2.0““.

В настоящото изследване ще бъдат дискутирани проблемите, с които се сблъскват незрящите потребители на някои широко разпространени среди за разработка на софтуер.

ИЗЛОЖЕНИЕ

Не е трудно да бъде забелязано, че докато за достъпността на продуктите на дигитализацията предназначени за масовия потребител се полагат определени усилия, достъпността на ориентирания към ИТ специалистите почти не се дискутира [3]. Аргументът, че хората с дефицити не избират подобни професии явно е несъстоятелен. Би следвало да се потърси обратната връзка – хората с увреждания рядко избират подобни професии, защото средствата, с които ще трябва да работят, са труднодостъпни. Все пак в някои страни работят незрящи и слабовиждащи ИТ специалисти, на които се налага да полагат значителни усилия и да изразходват немалко допълнително време при работа с необходимия хардуер и трудно достъпен за тях софтуер, което прави тези хора по-неконкурентноспособни на пазара на труда.

Ръководейки се от световните и европейските стандарти за достъпност [1] в споменатата област и изхождайки от опита си, авторът ще направи оценка на нивото на достъпност на някои често използвани среди за разработка на програмно осигуряване. Отправна точка ще бъдат препоръките на Европейския стандарт „Accessibility requirements for ICT products and services“ [1]. Ще бъде споделен опитът на автора за разработка на приложения на програмните езици C, C++, C# и Python с използване на някои от най-популярните среди за разработка. Ще бъдат споменати и тези от тях, които са напълно недостъпни. Особено внимание ще бъде обърнато на free ware средите за разработка и на тези, които се ползват преференциално от учащи се и учебни заведения, тъй като често те са първото специално „изпитание“, с което се сблъскват незрящите проявили интерес към професията „разработчик“ – изпитание, способно да мотивира, или демотивира.

1. Първа ще бъде разгледана средата за разработка DEV-C++. Това е софтуер с отворен код, предназначен за създаване на програмно осигуряване на C и C++. Предлагат се версии за MS Windows (от 95 до 10) и за Linux. Включени са средства за трасировка, debugger и за получаване на помощна информация. Целевият файл може да представлява приложение с графичен потребителски интерфейс (може да се използва MFC, което затруднява процеса на създаване на приложението), конзолно приложение, статична или динамична библиотека.

Навигацията по всички менюта и подменюта е напълно достъпна. Активирането на много от елементите им може да бъде извършено с клавиша <ENTER>, а на често използваните – с бързи клавиши. Навигацията в стандартните прозорци (например за отваряне и запис на файл) не предизвиква затруднения при работа с екранен четец. Предвидена е възможност за работа с отметки в кода (bookmarks), като навигацията в съответните прозорци не е свързана с трудности. В менюто “search” съществува елемент за преход на даден ред от текста на програмата, което облекчава процеса на коригиране на грешки (тъй като в диагностичните съобщения е посочен номерът на реда, на който е грешният код). За превключване между crr (респективно C) и съответния заглавен файл е предвиден бърз клавиш, което прави работата на разработчика не ползващ мишка много по-оперативна.

Изпълнението на основните действия в редактора на кода с помощта единствено на клавиатурата не е свързано със затруднения, маркирането, копирането, изтриването и преместването на текст се извършва с помощта на стандартните клавишни комбинации. Удобни за ползване са и прозорците за търсене и замяна на текст.

В повечето изскачащи прозорци не се озвучава текстът на съобщението, а само активните визуални обекти съдържащи се в тях (бутони “yes”/”no”, полета за отметка и т.н.), така потребител работещ с екранен четец не получава пълна информация за възникналата ситуация и е затруднен в преценката как да реагира. В прозореца за опции на проекта не се озвучават етикетите на текстовите и полетата за отметка.

По време на писането на кода средата извършва първична диагностика, като маркира със съответните цветове грешния код. Потребителят работещ с екранен четец не получава никаква информация за процеса на първична диагностика, той разбира за грешките в кода едва от резултатите на компилацията. Това прави работата му по-неекспедитивна. Генерираният при предикативното писане текст не се озвучава в реално време от екранния четец, което може да предизвика дублиране или пропускане на символи.

При работа без ползване на мишка липсва възможност за директен преход към областта с диагностичните съобщения на компилатора и свързващия редактор. Няма възможност също за бързо превключване между областите с текста на програмата и изведените при нейната работа резултати.

От казаното по-горе би могло да бъде направено заключение, че средата DEV-C++ е с ниска степен на достъпност в онези свои части, които са важни за комфорта и високата производителност на труда на разработчика служещ си с екранен четец. По ред причини тази среда за разработка, обаче, се използва широко в немалко висши учебни заведения и професионални гимназии. Ако в такава институция се обучават зрително затруднени използващи екранен четец, на преподавателите би следвало да се препоръча да разработват примери адаптирани за работа, както в средата DEV-C++, така и в друга с по-висока достъпност за зрително затруднени.

2. Може би най-разпространената среда за разработка в настоящия момент е “Microsoft visual studio”. Тя предлага възможност за разработка на приложен и базов софтуер на различни езици за програмиране, между които C#, C++, C, Python. Средата работи под управление на последните версии на Microsoft Windows, но е възможно създаването на приложения предназначени за използване и под управлението на други операционни системи. Възможна е разработката на конзолни и приложения с графичен потребителски интерфейс, създаването на статични и динамични библиотеки. Включени са средства за разработка на приложения ориентирани към работа с бази данни. Налични са средства за трасировка и debugging.

Навигацията по основното меню и подменютата, както и активирането на елементите им не е свързано със затруднения при работа с практически всички съвременни екранни четци. Същото може да бъде казано и за навигацията в повечето стандартни прозорци (отваряне и запис на файл, настройки). Пълноценно се озвучава текстът от изскачащите прозорци. Възможна е навигация в последните при използване на клавиатура. Много от често изпълняваните действия могат да бъдат извършени с активиране на клавишни комбинации.

В редактора на кода е възможно използването на стандартните клавишни комбинации за многоредово текстово поле, което осигурява базова достъпност. След формиране на необходимите навици, за незрящия разработчик се оказват лесни за използване средствата за предикативно писане, които повишават значително производителността му. Леснодостъпни са и средствата позволяващи автоматично генериране на шаблони за често използвани конструкции (например класове, условни оператори и цикли). Като повечето среди за разработка, “Microsoft visual studio” осигурява първична диагностика по време на писане на програмния код, индикирайки откритите грешки. Незрящият потребител практически не може да се възползва от очевидните предимства на тази функционалност, тъй като индикацията на грешките е реализирана с промяна на атрибутите на текста – нещо, което не се озвучава от екранните четци в реално време.

При създаване на програмен код на C#, C и C++ преминаването към областта с диагностичните съобщения на компилатора и свързващия редактор с използване на клавиатурата не предизвиква затруднения. Щракането с <ENTER> върху всяко от съобщенията за грешка при компилация предизвиква позициониране в мястото в кода на програмата, където е открита грешка. Подобно позициониране е по-сложно при работа на Python, главно заради появяването на диалогов прозорец, в който се предлага да се избере дали да се премине към изпълнение въпреки откритите грешки.

След обработката на програмния код се активира прозорецът визуализиращ работата на програмата на разработчика. В случаите, когато това е конзолно приложение, навигацията в този прозорец е същата като в команден прозорец. Ако

програмата е с графичен потребителски интерфейс, навигацията се осъществява в съответствие с предвиденото от програмиста.

При възникване на грешка по време на изпълнение на създадената програма, в горе споменатия прозорец се появява съответното съобщение, което като правило се озвучава от екранните четци. Не може да бъде определено като задоволително нивото на достъпност на средствата за трасировка и *debugging*. Въпреки предвидените бързи клавиши за поставяне на контролни точки и постъпково изпълнение, при работа с екранен четец проследяването на процеса на изпълнение на програмата е трудно, а наблюдаването на стойностите на избрани променливи е свързано със сложна навигация. Напълно недостъпни са средствата за диалогово създаване на графичен потребителски интерфейс. Същото може да се каже и за инструментите за създаване на приложения работещи с бази данни. Практически невъзможна е работата със системата за помощна информация.

С оглед казаното по-горе може да се твърди, че средата за разработка “Microsoft visual studio” може успешно да бъде използвана от потребители работещи с екранен четец за разработка на конзолни приложения. За *debugging* тези потребители следва да използват нестандартни подходи.

3. От средите за разработка на софтуер на “Borland” ще бъде разгледан “C++ builder 6”. Приложението работи под управление на версиите на MS Windows след 95. Създадените модули са предназначени за 32 битовите версии на MS Windows и Cylix. Могат да се създават: приложения с графичен потребителски интерфейс, конзолни приложения, модули за Контролния панел, модули реализиращи служби, динамични и статични библиотеки, ресурсни файлове и HTML документи. Възможно е използването на езиците за програмиране C, C++ и Assembler. Средата предлага средства ориентирани към работа с бази данни, графичен редактор, инструменти за трасировка и *debugging*.

Навигацията по всички менюта и подменюта може да се осъществява с помощта на клавиатурата, много от често използваните действия могат да бъдат активирани с клавишни комбинации. Съществува проблем с озвучаването на елементите на основното меню, които не се озвучават при навигация. Налага се да се използват клавишни комбинации, потребителите на “JAWS” могат да получават озвучена навигация по това меню с помощта на режима “JAWS cursor”. Практически всички прозорци осигуряват стандартна навигация с клавиатурата по включените в тях визуални компоненти. Последните са снабдени с текстови етикети, които се озвучават от екранните четци. При отваряне на някои изскачащи прозорци не се озвучава статичният текст.

В редактора на кода работят стандартните клавишни комбинации за многоредово текстово поле, което улеснява работата с екранни четци. За разлика от описаните по-рано среди за разработка, при навигация по редовете тук не се озвучават техните номера. Информацията за текущите ред и позиция е налична като текст и при

използване на “JAWS” може да бъде озвучавана с клавишна комбинация при създаване на съответния скрипт. Озвучаването при предикативното писане се свежда до показване на уточнените имена на структурите и класовете. Изскачащият текст се озвучава ефективно в реално време.

Преминаването от кода на програмата към областта с диагностичните съобщения на компилатора и свързващия редактор, както и към прозореца, в който работи потребителската програма, не предизвиква затруднения. Диагностичните съобщения се озвучават безпроблемно. При щракване с клавиша <ENTER> върху съобщението за грешка курсорът се премества на мястото в кода на програмата, където е съответната грешка.

Средствата за debugging са достъпни и позволяват на работещият с екранен четец ефективно да следи хода на изпълнение на своята програма и да наблюдава стойностите на избраните променливи в контролните точки. Достъпни при работа с “JAWS” са и средствата ориентирани към работа с бази данни, на определени места се налага да се работи в режим “JAWS cursor”. Недостъпни остават важните функции на графичния редактор.

За разлика от описаните по-горе среди за разработка “C++ Builder 6” позволява на незрящия потребител работещ с “JAWS” да създава в диалогов режим приложения с графичен потребителски интерфейс – достъпни са прозорците на палитрата с предлаганите компоненти (визуални и невизуални), прозорецът съдържащ дървото на компонентите на приложението и прозорецът за настройка на свойствата на компонентите (Object inspector). Компонентите (полета за отметка, радиобутони и т.н.) в съответната палитра са снабдени с текстови етикети, които се озвучават или извеждат на брайлов дисплей. С помощта на стандартен бутон избраният компонент се премества в панела за конфигуриране на прозореца, където продължава да е снабден с текстов етикет. Управлението на атрибутите на компонентите поместени в прозореца за конфигуриране се осъществява с помощта на специален модул на средата за разработка – “Object inspector”, който е реализиран като стандартен прозорец и осигурява практически пълна достъпност. С негова помощ се задават свойствата и разположението на компонентите в диалоговите прозорци, както и реакцията на генерираните автоматично модули на определени събития. Следва да се отбележи, че по подобен начин се управлява създаването и настройката не само на чисто визуалните компоненти (бутони, полета за отметка и т.п.), палитрата на компонентите съдържа и такива ориентирани към работа с бази данни, мрежов трансфер, работа със звук и др. Високата степен на достъпност на вградената система за помощна информация увеличава значително производителността на труда на незрящия разработчик.

Немаловажен е фактът, че визуалните компоненти, включени в библиотеките на “C++ Builder 6”, осигуряват висока степен на достъпност на създаденото приложение.

От казаното по-горе следва, че “C++ Builder 6” осигурява висока степен на достъпност при работа с екранния четец “JAWS” при разработка на широк кръг

приложения. Това вероятно е най-всестранно достъпната среда за разработка до момента. За съжаление, след организационните и структурни промени в “Borland” наследниците на тази среда за разработка се оказват все по-трудно достъпни за незрящия програмист. Не съвсем ясен е пътят за евентуалното лицензиране на продукта в настоящия момент. Въпреки двадесетина годишната си възраст “C++ Builder” все още може да бъде полезен при разработка на 32 битови приложения за РС, но основната му ценност в настоящия момент е да служи като аргумент в дискусията с компаниите разработващи подобен софтуер, аргумент в полза на твърдението, че изцяло достъпни среди за разработка може и трябва да има.

РЕЗУЛТАТИ

В таблица 1 по-долу са дадени измерените времена в часове и минути за разработка от незрящ C++ програмист на 3 приложения с използване на “DEV-C++”, “Microsoft visual studio” и “C++ Builder 6”. Първото приложение е с най-ниска сложност и представлява реализация на бинарно търсене в сортиран масив. Второто е с по-висока сложност и е реализация на алгоритъма на Дейкстра за намиране на най-краткия път от даден връх до всеки един от останалите в граф с неотрицателни тегла, третото приложение - несложен текстов редактор подобен на “Notepad”.

Таблица 1

Приложение	DEV-C++ (час:минути)	Microsoft visual studio (час:минути)	C++ Builder 6 (час:минути)
Бинарно търсене в сортиран масив	0:30	0:14	0:15
Алгоритъм на Дейкстра	2:42	1:59	1:57
Несложен текстов редактор	-	-	3:01

Разбира се, това е само единичен експеримент, поради липса на незрящи програмисти нямаше как да бъде формирана съвкупност с по-голяма представителност, но според автора тези резултати са близки до реалните съотношения на разход на време за разработка на софтуер.

Няма основание да се предполага, че тенденцията за силна визуална ориентация на средите за разработка на програмно осигуряване, а и на софтуера ориентиран към системните администратори и операторите на сложни системи за управление, ще

отслабва. Графичният интерфейс носи много удобства за зрящия потребител. В същото време прави по-сложна задачата на екранните четци. Това обстоятелство насочва усилията на разработчиците на средства улесняващи работата с компютър на хората със зрителен дефицит към създаване на специализирани операционни среди, а дори и на версии на операционните системи адаптирани за незрящи. Пример за такава система е Luwrain [4, 5] - разработена от Доцент Д-р Михаил Пожидаев – незрящ преподавател по информационни технологии в Томския държавен университет. Luwrain може да е част от ядрото на Linux, или платформено независима операционна среда. Системата осигурява удобен интерфейс за зрително затруднения потребител, позволявайки му в същото време във всеки един момент да превключи към работа с избрания екранен четец. По този начин Luwrain не е алтернатива на екранните четци, а мощно допълнително средство за достъп повишаващо ефективността на работата на незрящия потребител.

ЗАКЛЮЧЕНИЕ

В заключение може да се направи изводът, че нивото на достъпност на софтуера за разработка на програмно осигуряване е на по-ниско ниво от достъпността на дигиталните средства ориентирани към широкия потребител. Това обстоятелство ограничава незрящите младежи в избора на професия, а последното, съгласно съвременните разбиращения следва да бъде определено като дискриминация.

ФИНАНСИРАНЕ

Изследването е частично финансирано по проект „Дигитална достъпност за хора със специални потребности: методология, концептуални модели и иновативни екосистеми“ чрез договор №КП-06-Н42/4 от 08.12.2020 г. с фонд „Научни изследвания“ на Република България.

ЛИТЕРАТУРА

1. Accessibility requirements for ICT products and services, <https://joinup.ec.europa.eu/collection/rolling-plan-ict-standardisation/accessibility-ict-products-and-services>
2. Negoslav Sabeв, 2020, „Computer methods and approaches by investigation and representation of knowledge for WEB accessibility for disabled people“. Institute of mathematics and informatics, BAS.
3. Negoslav Sabeв, Galya Georgieva-Tsaneва, Galina Bogdanova, 2018, „Creating a Software System with Functionality to Help Make it Accessible for People with a Visual Deficit“, CBU International Conference Proceedings, Vol. 6, pp. 734-738.
4. <http://luwrain.org/index.php?lang=en&mode=normal>
5. Mikhail Pozhidaev, 2015, „Introducing LUWRAIN: Can GNU/Linux help us rethink accessibility solutions for the blind?“, Linux Journal, №7, Texas, USA.
6. Galya Georgieva-Tsaneва, Negoslav Sabeв, 2018, „Technologies, Standards and Approaches to Ensure Web Accessibility for Visually Impaired People“, Digital Preservation and Presentation of Cultural and Scientific Heritage, Conference Proceedings. Vol. 8, Sofia, Bulgaria: Institute of Mathematics and Informatics – BAS, 2018. ISSN: 1314-4006, eISSN: 2535-0366, pp. 143-150.