

## GRAPHIC USER INTERFACE

Vinnitsia National Technical University

### Анотація

*В даній роботі було досліджено значимість графічного інтерфейсу користувача, види існуючих платформ для конкретних операційних систем. Також висвітлено історію виникнення графічних інтерфейсів та їх перші застосування*

**Ключові слова:** *графічний інтерфейс користувача, дослідження, інтерфейс, ОС, Windows, Linux, Qt, C++, C#.*

### Abstract

*The article deals with the importance of a graphical user interface, the types of existing platforms for specific operating systems. Also the history of graphic interfaces and their first applications are presented.*

**Keywords:** *GUI, research, interface, OS, Windows, Linux, Qt, C++, C#.*

Due to the research conducted at Stanford Research Institute in the 60s of last century, Doug Engelbart invented the graphical user interface.

Subsequently researchers from the Xerox PARC research lab adopted GUI concept in the 1970s. In 1973, Xerox PARC lab gathered young scientists and the research was given freedom. As a result, among other things, the concept of graphical interface WIMP (Windows, Icons, Menus, Point-n-Click) appeared and within the framework of this concept Alto computer was created. It was not released as a commercial product, but it was widely used in the company as a corporate tool Xerox.

The commercial embodiment of the GUI concept was received in 1984 as Apple Computer products. Currently, GUI is a standard component of most available operating systems and applications on the market. Although the vast majority of GUI system is a superset of the operating system, there are also independent of its implementation. The known version of BIOS Setup graphics program was similar to full GUI. However, this BIOS option did not pass the test of time.

The graphical user interface is a system of tools for user interaction with the device based on the idea of user-accessible system objects and functions as a graphical display components (windows, buttons, scroll bars, and so on). When working with a GUI user has a random access (via the keyboard or the coordinate input device, for example, a touch-screen) to all visible display objects. The first graphical user interface was implemented in the operating systems of personal computers but now the GUI elements have become an integral part of even simple household and medical devices, smartphones, industrial automation, and many others.

According to the classifications there are following types of GUI:

- simple: standard screen forms and standard user interface elements provided by the GUI subsystem itself;
- true graphics, two-dimensional: non-standard interface elements and original metaphors to realize their own application with the library or a third party;
- three-dimensional.

One of the requirements for a good graphical user interface of a software system is the concept of Do What I Mean or DWIM. DWIM requires the system to work predictably; the user intuitively knows in advance what action the program does after obtaining his command.

The advantages include:

- the GUI that is "friendly" for users who have started acquaintance to a computer with a graphical interface;
- the graphics processing programs, it is often the only possible.

Moreover, there are disadvantages. They are:

- large memory consumption in compared to a text-based interface;
- more difficult remote work;
- inability of automation in some cases;
- the graphical interface that is not "friendly" for users who have started acquaintance to a computer with the command line interface.

Let us consider the most popular at present ways to build interface client applications in C ++ as the most used for the development of software for OS Microsoft Windows (MS Windows) and Linux. The main way of software development for MS Windows is MS Visual Studio. This integrated development environment (IDE) enables the development of software in different programming languages but the main languages, of course, are C ++ and C #. If we talk about low-budget projects, the most suitable option is the Linux operating system. Apart from the fact that most of the distributions of this OS are free including commercial use, there are a number of free tools for developing quality software for Linux. The most common tool for software development in C++ is a cross-platform toolkit Qt. It is important to emphasize that Qt lets you develop applications not only for Linux but also for MS Windows, Mac OS X, Android, and other UNIX-like operating systems.

An analogy to MS Visual Studio is Qt IDE Qt Creator. The alternative of WinForms are so-called widgets (Qt Widgets) as an alternative to WPF - Qt Quick. Also in Qt Creator you can create HTML5-based interfaces. However, the most interesting toolkit module is embedded web engine WebKit, which is the basis of all modern web browsers. This module is available in MS Visual Studio but it has some limitations. Qt WebKit module allows you to create a client application interface using the interface technology development of web applications. A well-established technology stack is at the heart of creating a web application interface. It includes the markup language HTML (HTML 4, 5), Cascading Style Sheets (CSS 2, 3) and JavaScript scripting language with a rich selection of additional libraries (skeletons). Special attention is given to the fact that the rate of appearance of new useful scaffolds for the JavaScript language is growing rapidly, and this makes the design and functionality more quick and easy.

Analyzing the abovementioned information, we can state the experimental research proves the importance of GUI in today`s life. The following methods of GUI creation presented in the research are actual and the most popular for today.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ / REFERENCES

1. Халитов Кирилл Универсальный GUI / Кирилл Халитов. – [Електронний ресурс] – [Режим доступу] <https://habrahabr.ru/post/244281/>.
2. Електронна бібліотека Вікіпедія [Електронний ресурс] UML – Вікіпедія – [Режим доступу] <http://ru.wikipedia.org/wiki/>
3. Калиниченко Л. А. Язык определения интерфейсов IDL в архитектуре / Л. А. Калиниченко, М. Р. Когаловский – Москва : Вильямс. – 2005.

*Науковий керівник: Тульчак Людмила Володимирівна – старший викладач кафедри іноземних мов, Вінницький національний технічний університет*

*Булига Андрій Ігорович – студент групи 1АВ-14б, факультет комп'ютерних систем та автоматики, Вінницький національний технічний університет, м. Вінниця, [anbulyga@rambler.ru](mailto:anbulyga@rambler.ru)*

*Supervisor: Liudmyla Tulchak – senior teacher of English, the Foreign Languages Department, Vinnytsia National Technical University, Vinnytsia*

*Andrii Bulyha, group 1AV-14, Faculty of Computer Systems and Automatics, Vinnytsia National Technical University, Vinnytsia.*