DOI: 10.36910/6775-2524-0560-2020-40-16
УДК: 004.4
[1]**Глинчук Людмила Ярославівна**, к. ф.-м. н., доцент
https://orcid.org/0000-0002-8943-9604
[1]**Гришанович Тетяна Олександрівна**, к. ф.-м. н., ст. викладач
https://orcid.org/0000-0002-3595-6964
[2]**Кузьмич Олена Іванівна**, к. ф.-м. н., доцент
https://orcid.org/0000-0002-8717-4497
[2]**Багнюк Наталія Володимирівна**, к. т. н., доцент
https://orcid.org/0000-0002-7120-5455
[1]Східноєвропейський національний університет імені Лесі Українки, м. Луцьк, Україна
[2]Луцький національний технічний університет, м. Луцьк, Україна

ЕФЕКТИВНЕ ВИКОРИСТАННЯ НОВІТНІХ МЕТОДІВ ПРОГРАМУВАННЯ ГРАФІКИ НА С++ В НАВЧАЛЬНИХ ЦІЛЯХ

Глинчук Л.Я., Гришанович Т.О., Кузьмич О.І., Багнюк Н.В. Ефективне використання новітніх методів програмування графіки на С++ в навчальних цілях. Для того щоб вміти програмувати та налаштовувати графіку з використанням сучасних бібліотек, фреймворків і т.д. під різні операційні системи, потрібно знати як класичний інструментарій, так і новітні методи. При програмуванні, зокрема на С++, та налаштуванні різних додатків (середовищ програмування) і підключенні графічних бібліотек під певну ОС виникають деякі труднощі та проблеми. Саме тому повинне бути розуміння щодо сумісності та можливого вирішення такої проблеми. З цією метою в даній статті розглядаються вибрані графічні бібліотеки - від примітивних до сучасних та описуються особливості роботи з ними.
Ключові слова: програмування, налаштування, графічна бібліотека, мова програмування С++.

Глинчук Л.Я., Гришанович Т.О., Кузмич Е.И., Багнюк Н.В. Эффективное использование новых методов программирования графики на С++ в учебных целях. Для того, чтобы уметь программировать и настраивать графику с использованием современных библиотек, фреймворков под разные ОС, нужно знать как классический инструментарий, так и новые методы. При программировании, в частности на С++, и настройке разных приложений (среды программирования) и подключении графических библиотек под определенную ОС возникают определенные трудности и проблемы, потому должно быть понимание. Именно поэтому должно быть понимание того касательно совместимости и возможного решения этой проблемы. С этой целью и рассматриваются некоторые графические библиотеки - от примитивной к современной и описываются особенности работы с ними.
Ключевые слова: программирование, настройка, графическая библиотека, язык программирования С++.

Hlynchuk L. Y., Hryshanovych T. O., Kuzmych E., Bahniuk N. Effective using of programming and tuning methods of graphic on C++ for educational aims. In order to program and influence the graphic arts with the use of modern libraries, frameworks under different OS, it is needed to know classical and newest methods. At programming, in particular on C++, and tuning of different additions (compilers) and connecting of graphic libraries under certain OS there are some difficulties and problems, that is why there must be understanding: compatible or not compatible and why; whether it is possible to do so that it was compatible. To that end and some graphic libraries are examined: from primitive to modern and the features of work are described with them. **Keywords:** programming, tuning, graphic library, programming of C++ language

Постановка проблеми та аналіз досліджень. Насьогодні є достатня кількість на ринку операційних систем (ОС) для комфортної та продуктивної роботи в галузі інформаційних технологій. Але створення програмованої графіки все ще залишається непростим завданням, оскільки для такого роду задач слід використовувати специфічні засоби. Загалом ідея програмування графіки зводиться до двох напрямів: або використовувати засоби системи (для операційної системи сімейства Windows це, наприклад, WinAPI), або за допомогою призначених для цього бібліотек та фреймворків.

В цьому контексті розглянемо насамперед власне мову програмування С++. Оскільки це - мова високого рівня, яка підтримує декілька парадигм програмування та активно розробляється (версія С++20 має з'явитися у поточному 2020 році), то логічно, що і створювати графічні зображення із використанням її засобів можна. Проте під час детального аналізу літератури, присвяченої технології програмування мовою С++, помітно, що у багатьох авторів, зокрема, у творця мови С++ Страуструпа Бьерна у завершеному 2-му виданні [1] не має детального опису програмування графіки. У Навроцкого А. А. [2] теж немає. Складається враження, що мова програмування і власне сама графіка використовується незалежно, але це не зовсім так. Причина скоріше в тому, що надбудов для програмування графіки вистачає, але їх при необхідності потрібно налаштовувати та додатково прописувати. Тому і більшість літератури по С++ містить загальну основну структуру, але є і автори, які розглядають графіку та дають можливість навчитися будувати графічні програми. Наприклад, розробник цілої низки навчальних посібників "Програмування. Python, С++" для 8-11 класів, Поляков

© Глинчук Л.Я., Гришанович Т.О., Кузьмич О.І., Багнюк Н.В

К. Ю., д.т.н., вчитель вищої категорії, якраз і описує налаштування графіки на своєму сайті [3]. Також Глинський Я. М., Анохін В. Є., Ряжська В. А. у своєму посібнику [4] теж описують побудову графічних програм, наводять графічні функції, проте не описують налаштувань. Автор більше 1000 вихідних текстів, детальних роз'яснень та кодів, Д. Семенидо у своїй статті [5] дуже детально та зрозуміло описує, як можна малювати використовуючи мову С++, є навіть приклади, які працюють вірно та без помилок. Як програмувати графіку використовуючи WinApi – питання проаналізоване у [8], [9], [5] та ін. Поступово переходячи від простих бібліотек до складніших, розглянемо ще графічну бібліотеку OpenGL. Робота з нею описана у багатьох авторів, зокрема українськими авторами у [9, 10] та багато іншими. Розуміння питання програмування графіки можна проаналізувати також на форумах, проте там не завжди є цілісне бачення.

Метою досліджень є аналіз особливостей технології програмування та налаштування графіки на мові С++ з використанням середовищ програмування у консольному режимі.

Виклад основного матеріалу. Отож, у [5] автор пояснює програмування графіки починаючи від ОС DOS. У такому випадку програмістам було нелегко, оскільки роботу з графікою доводилося виконувати напряму, тому і з'явилися графічні бібліотеки. Однією з перших була графічна бібліотека компанії Borland – Borland Graphics Interface (BGI). Усі середовища програмування на С++ під DOS використовували дану бібліотеку. Останнє середовище програмування, що підтримувало BGI було Borland C++ 5.02, хоча воно працювало уже під ОС Windows, проте мало функціонал для компіляції програм під DOS через бібліотеку graphics.lib шляхом підключення заголовочного файлу *graphics.h.* [6]

Робота з графікою ускладнювалася ще і тим, що потрібно було підключати спеціальний драйвер для ініціалізації та роботи з графічним режимом відеоплати. У [3] описано налаштування графіки для оболонки Dev-C++ з використанням заголовочного файлу *graphics.h* та бібліотеки *libbgi.a.* Але для ОС Windows 8/10 оболонка Dev-C++ уже не працює, проте є сучасна версія, яка налаштовується якраз під роботу вказаних ОС. Відмінність полягає лише в тому, що тепер є потрібними два заголовочні файли *graphics.h* та *winbgim.h* і бібліотеку *libbgi.a*. У відеоматеріалі [13] детально показано налаштування. Аналогічних відео вистачає, тому при розумінні загальної картинки все стає на місця.

Особливістю бібліотеки graphics є її простота у використанні, тому для початківця вона дуже зручна. Маємо перелік функцій (до прикладу, у джерелі [14]), для виконання операції використовуємо функції з відповідними параметрами, при цьому не потрібно використовувати ніяких складних структур і т.д. Проте мінусом є можливість використання всього 16 кольорів. Але якщо ви налаштовуєте графіку ще і з файлом *winbgim*, то це багато що міняє. Що ж таке *Winbgim*? Для того, щоб була змога програмувати графіку в ОС Windows 8/10, Міхаель Майн допрацював графічну бібліотеку під компілятор MinGW, додавши туди все необхідне і надавши можливість використовувати уже всю палітру кольорів RGB. Плюси: легко підключати, зручно використовувати, не вимагає значних ресурсів системи, працює з будь-якими розширеннями та адаптерами, є можливість програмувати дії для мишки та клавіатури. Мінусів небагато: для виведення тексту у графічному режимі немає великого набору шрифтів, невеликий набір заливок та ліній. [7] Функції цієї бібліотеки можна переглянути на сайті [15]. Додаткові функції позначені позначкою win. Тобто Winbgim – так званий порт BGI для Windows. Аналогічний процес налаштування та програмування графіки і для вільного багатоплатформного середовища розробки Code::Blocks. Розглянемо детально процес налаштування та програмування графіки у Code::Blocks версії 17.12.

Для того, щоб налаштувати графічний режим з використанням *graphics.h, winbgim.h* та бібліотеки *libbgi.a,* потрібно зробити декілька кроків з перенесенням та підключенням необхідних файлів:

1. З джерела [16] завантажити та розпакувати Graphics-Library-master (рис.1).

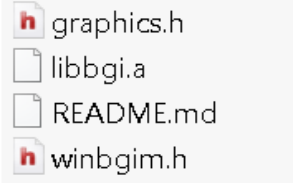

Рис. 1. Файли, що містяться в завантаженому архіві

2. Розмістити файли у відповідні папки, тобто:

a) Скопіювати та вставити файли graphics.h и winbgim.h в папку include директорії Code::Blocks. Шлях: C: \ Program Files (x86) \ CodeBlocks \ MinGW \ include.

b) Скопіювати і вставити файл libbgi.a в папку lib в Code: Blocks.

© Глинчук Л.Я., Гришанович Т.О., Кузьмич О.І., Багнюк Н.В

Шлях: C: \ Program Files (x86) \ CodeBlocks \ MinGW \ lib.

3. Також необхідно в додатку Code::Blocks, перейти у вкладку Налаштування/ Компілятор, тобто як на рис. 2.

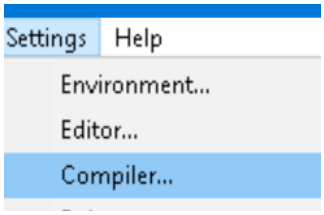

Рис. 2. Вкладка Налаштування/Компілятор

Далі потрібно перейти на вкладку Linker Settings. В Link Libraries додайте та перейдіть до C: \ Program Files (x86) \ CodeBlocks \ MinGW \ lib \ та виберіть libbgi.a.

Вставити наступний текст у вкладку Other linker options (тобто з правої сторони)

-lbgi -lgdi32 -lcomdlg32 -luuid -loleaut32 -lole32.

Після попередніх дій картинка повинна мати вигляд як на рис. 3. Після таких кроків потрібно зберегти налаштування та перезавантажити додаток. Якщо файли graphics.h, winbgim.h, libbgi.a не старої версії, то після запуску помилок не повинно виникати. Після налаштувань необхідно перевірити, чи все працює правильно, тому запускаємо Code::Blocks і вставляємо в нього заготовлений наступний код:

```
#include <graphics.h>
#include <stdio.h>
#include <conio.h>
main ()
{ int gdriver = DETECT, gmode, errorcode;
// Задання графічного режиму
initgraph(&gdriver, &gmode, "");
// --- щось малюємо ---
getch();
closegraph();
return 0;}
```

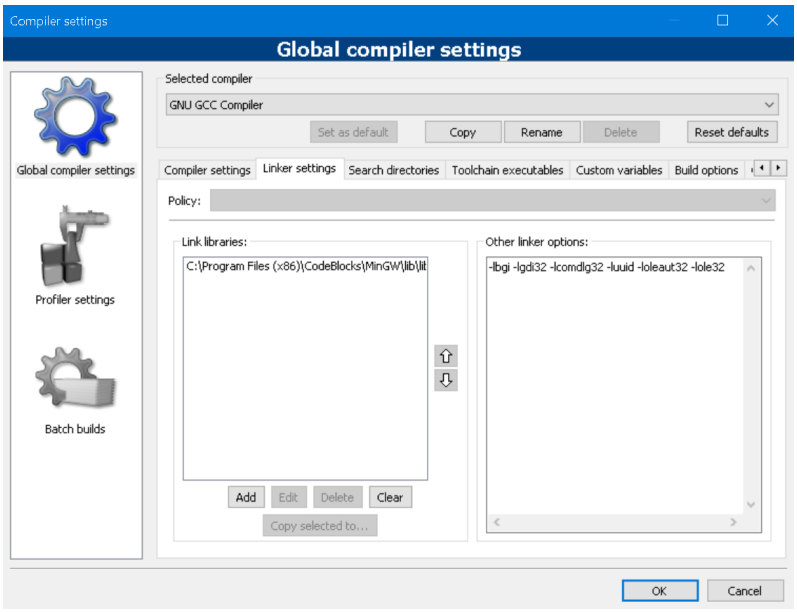

Рис. 3. Налаштування Linker Settings

Якщо графічний режим налаштовано правильно, то після виконання (компіляції) даного коду повинно запуститися 2 чорних екрани: один Windows BGI, інший – назва файлу.exe. Для малювання необхідних вам елементів у блоці // --- щось малюємо --- , можна використовувати усі графічні функції

© Глинчук Л.Я., Гришанович Т.О., Кузьмич О.І., Багнюк Н.В

бібліотеки graphics.h, до прикладу, такий код намалює зафарбований прямокутник та зафарбоване коло:

```
setcolor(12);
rectangle(120,100,400,300);
setcolor(14);
bar(120,100,400,300);
setcolor(13);
setfillstyle(1,13);
circle(400,300,100);
floodfill(400,300,13);
```

Ідея програмування графіки у С++ подібна до так званого принципу "сендвіча", тобто спочатку потрібно проініціалізувати графіку (задати графічний режим), а далі можемо малювати, і нарешті, закрити графічний режим. В даному випадку ініціалізує графіку функція initgraph(), яка має 3 вхідних параметри: перший – вказує тип відеоадаптера, другий – графічний режим, тобто розширення, третій – шлях до файлу драйвера; закриває графічний режим функція closegraph(). Така графічні бібліотеки дозволяють малювати лише 2D графіку.

Розглянемо бібліотеку *windows.h* (спеціально розроблений для Windows, заголовочний файл, для мов програмування С та С++), однією з можливостей якої є програмування графіки на С++ із використанням графічних команд WinApi. Все дуже просто у налаштуванні, ніякі файли нікуди не потрібно переміщати, нічого не потрібно прописувати, але сам шаблон програми дещо складніший, ніж у попередньому прикладі. Програма для Win 32 зазвичай складається з таких блоків як показано у [9] п.4. На початку потрібно підключити бібліотеку, тобто #include <windows.h> далі згідно блоків вписувати код. Програмування графіки тут відбувається за аналогією: є головна функція, проте запис інший; є блок ініціалізації; є блок, де починають малювання, він у свою чергу складається з функцій, коли вказують початок малювання, описують самі функції малювання та функцію кінця програмування. Як приклад розглянемо частину коду:

```
#include<windows.h>
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpszCmdLine, int nCmdShow)
{…
//повідомлення про малювання
case WM_PAINT :
hdc=BeginPaint(hWnd, &ps); //початок малювання
//малюємо заокруглений прямокутник
hBrush=CreateSolidBrush(RGB(250,200,100));
SelectObject(hdc, hBrush);
hPen=CreatePen(2,2,RGB(0,0,255));
SelectObject(hdc, hPen);
RoundRect(hdc, 20, 250, 250, 350, 15, 15);
ValidateRect(hWnd, NULL); // обновляємо вікно
EndPaint(hWnd, &ps); //кінець малювання
break;
…}
```

І це ще не весь код, а тільки та частина, яка відповідає за процес прорисовки. Насправді рядків коду набагато більше у порівнянні з попереднім прикладом і функції дещо складніші для розуміння. Сам шаблон такої графічної програми теж громіздкий і містить декілька блоків, які в свою чергу дозволяють зробити усі необхідні описи, бо, пропустивши ту чи іншу частину, програма уже не працюватиме. Використання windows.h дозволяє малювати 2D графіку, але має набагато більше можливостей, бо "знає все, що знає Windows".

І, нарешті, розглянемо потужну графічне бібліотеке OpenGL. Це один з найпопулярніших кросплатформних API для розробки 2D та 3D графічних додатків. Крім того, ця бібліотека відкрита, незалежна від мови програмування та налічує більше 250 функцій, які можна використовувати при програмуванні графіки. Та і популярна тим, що має велику кількість якісної документації та прикладів. Розглянемо детальніше основні ідеї. OpenGL складається з набору бібліотек: GLU, GL, GLUT, GLX, кожна з яких має свої функції. Функції OpenGL працюють у вигляді моделі клієнт-сервер. Додаток виробляє команди та виступає в ролі клієнта, а OpenGL виступає сервером, тобто інтерпретує та

© Глинчук Л.Я., Гришанович Т.О., Кузьмич О.І., Багнюк Н.В

виконує їх. Якщо використовувати бібліотеку GL, то, наприклад, у середовищі програмування Code::Blocks вона є в явному вигляді. Якщо потрібно використати бібліотеку GLUT та мати можливість використовувати її функції, потрібно аналогічно, як і для graphics.h, здійснити налаштування. По аналогії помістити певні файли у відповідні папки. Детально робоче (перевірене) налаштування описане у [12]. Для того, аби перевірити чи ми зможемо програмувати графіку потрібно зайти у File>New>Project… вибрати Glut project і далі за інструкцією, або File>New>Project… вибрати Console application і у головне вікно вставити наступний код:

```
#include <windows.h>
#include <GL/glut.h> // підключаємо бібліотеки
void Drow()
{// щось малюємо }
int main(int argc, char *argv[])
{  glutInit(&argc, argv);  // ініціалізація графіки стандартно
   glutInitWindowSize(640,480); // задаємо розміри вікна
   glutInitWindowPosition(20,10); // задаємо позицію, з якої буде виводитися вікно
   glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH); // задаємо кольори дисплею
   glutCreateWindow("GLUT Shapes"); // задаємо назву графічного вікна
   glutDisplayFunc(Drow); // щось малюємо
   glutMainLoop(); } // функція відповідає за обробку подій
```

В результаті виконання з'явиться 2 вікна: одне – назва.exe, інше – створено програмно, воно буде мати біле дно і назву таку як задали у функції glutCreateWindow("GLUT Shapes"). Особливістю програмування графіки є те, що усі функції бібліотек мають префікс назви бібліотеки, в даному випадку glut.

Наведемо приклад програмування побудови трикутника з різними кольорами у вершинах:

```
static void Drow(void)
{   glColor3d(1,0,0);
   glBegin (GL_TRIANGLES);
   glColor3f (12.0, 0.0, 0.0); //червоний
   glVertex3f (0.0, 0.0, 0.0);
   glColor3f(0,1,0); //зелений
   glVertex3f (1.0, 0.0, 0.0);
   glColor3f(0.0,0.0,1.0); //блакитний
   glVertex3f (1.0, 1.0, 0.0);
   glEnd ();
   glFlush (); }
```

Команди мають префікс gl, тому так програмувати можна, оскільки на початку програми підключено одразу 2 бібліотеки #include <GL/glut.h>. Звертаємо увагу на те, що і тут застосовано принцип "сендвіча": є glBegin(), сама побудова фігури та glEnd (). Таким чином, вивчаючи детально функції бібліотеки OpenGL можна програмувати не тільки 2D, а і професійну 3D графіку. На даний момент існує багато графічних надбудов на C++, тому головне зрозуміти техніку такого програмування. Отож, під час вивчення графіки на C++ зі студентами потрібно починати з самої простої або як ще її називають "примітивної графіки для лінивих" [7] та поступово перейти до складніших. Таким чином буде на практиці відчутно, які методи для розуміння є простішими та легшими у використанні, а які є складнішими, але дають багато можливостей.

Висновки. Розглянуто приклади налаштування та програмування графіки на C++ з використанням різних графічних бібліотек та різних середовищ програмування під ОС Windows. Однак, потрібно завжди враховувати особливості тієї чи іншої технології, середовища програмування, компілятора та ОС. Необхідно уважно вивчати всі деталі на сумісність і можливість об'єднання їх при створені графічних зображень. Розглядаючи детально дане питання, стало зрозуміло, що для мови програмування C++ вистачає різноманітних надбудов для роботи з графікою. Для того, щоб ефективно програмувати, в тому числі навчальні проекти, використовуючи графіку в правильному напрямку, необхідно виписати список функцій, які планується використовувати при програмуванні проєкту; вивчити всю необхідну в даному напрямі офіційну документацію мови програмування; вивчити опис усіх необхідних бібліотек та фреймворків та їх особливостей використання. І тільки після детального

© Глинчук Л.Я., Гришанович Т.О., Кузьмич О.І., Багнюк Н.В

вивчення та аналізу потрібно приймати рішення про вибір програмного інструментарію, який дійсно найбільше підходить для реалізації проєкту.

Список бібліографічного опису

1. Страуструп Бьерн. Язык программирования C++ для профессионалов. 2-е изд. — М.: Интуит, 2016. — 670 с.
2. Навроцкий, А. А. Основы алгоритмизации и программирования в среде Visual C++ : учеб.-метод. пособие / А. А. Навроцкий. – Минск : БГУИР, 2014. – 160 с.: ил.
3. Как писать простые графические программы? Как работать с графикой в Windows 8/10? [Електронний ресурс] — Режим доступу: http://kpolyakov.spb.ru/school/c/faq.htm#bgi
4. Глинський Я. М., Анохін В. Є., Ряжська В. А. C++ і C++ Builder. – Львів: Деол, СПД Глинський, 2003. – 192 с.
5. Семенидо Д. Как программировать на C++ для DOS и Windows. [Електронний ресурс] — Режим доступу: http://radiofront.narod.ru/htm/prog/htm/indexprog.html
6. Borland Graphics Interface. [Електронний ресурс] — Режим доступу: https://ru.wikipedia.org/wiki/Borland_Graphics_Interface
7. Примитивная графика. 2019 р. [Електронний ресурс] — Режим доступу: https://habr.com/ru/post/455056/
8. Вежневець В. Краткое неформальное введение в графику Windows. [Електронний ресурс] — Режим доступу: https://www.graphicon.ru/oldgr/courses/cg/library/windows/index.html
9. Соловйов О. Основы программирования для Win32 Api. [Електронний ресурс] — Режим доступу: https://dims.karelia.ru/win32/
10. Лященко А.А., Демченко В.В., Бородавка Є.В. Геометричне моделювання і комп'ютерна графіка: використання бібліотеки OpenGL: Навчальний посібник. — К.: КНУБА, 2008. — 76 с.
11. Свірневський М. С., Ковальчук С. С. Основи розробки графічних додатків: Навчальний посібник – Хмельницький: ХНУ, 2015. – 260 с.
12. Using OpenGL & GLUT in Code::Blocks. [Електронний ресурс] — Режим доступу: http://www.sci.brooklyn.cuny.edu/~goetz/codeblocks/glut/
13. How to Make Graphics in Dev C++ on Windows 10 [Електронний ресурс] — Режим доступу: https://www.youtube.com/watch?v=DcsAUO2QDqc
14. Справочник по C/C++. Графический режим. [Електронний ресурс] — Режим доступу: http://mycpp.ru/cpp/scpp/cppd_graphics.h.htm
15. Borland Graphics Interface (BGI) for Windows. [Електронний ресурс] — Режим доступу: https://www.cs.colorado.edu/~main/cs1300/doc/bgi/
16. Graphics-Library. [Електронний ресурс] — Режим доступу: https://github.com/SagarGaniga/Graphics-Library
17. V.Satsyk R.Grudetsky, O.Kuzmych, N.Bahniuk, L.Hlynchuk Y.Melnychuk Reduction of Server Load by Means of CMS Drupal. IEEExplore Digital Library, Published in: 2020 10th International Conference on Advanced Computer Information Technologies, Deggendorf, 16-18 September 2020.

References

1. Stroustrup Björn. C ++ programming language for professionals. 2nd ed. - M .: Intuit, 2016 .-- 670 p.
2. Navrotsky, A. A. Fundamentals of algorithms and programming in the Visual C ++ environment: textbook. allowance / A.A. Navrotsky. - Minsk: BSUIR, 2014 -- 160 p .
3. How to write simple graphics programs? How to work with graphics in Windows 8/10? [Electronic resource] - Access mode: http://kpolyakov.spb.ru/school/c/faq.htm#bgi
4. Glinsky Ya.M., Anokhin V.Y., Ryazhka V.A. C ++ i C ++ Builder. - Lviv: Deol, SPD Glinsky, 2003 .-- 192 p.
5. Semenido D. How to program in C ++ for DOS and Windows. [Electronic resource] - Access mode: http://radiofront.narod.ru/htm/prog/htm/indexprog.html
6. Borland Graphics Interface. [Electronic resource] - Access mode: https://ru.wikipedia.org/wiki/Borland_Graphics_Interface
7. Primitive graphics for lazy, oldfag and lazy oldfag. 2019 p. [Electronic resource] - Access mode: https://habr.com/ru/post/455056/
8. Vezhnevets V. A short informal introduction to Windows graphics. [Electronic resource] - Access mode: https://www.graphicon.ru/oldgr/courses/cg/library/windows/index.html
9. Solovyov O. Fundamentals of programming for Win32 Api. [Electronic resource] - Access mode: https://dims.karelia.ru/win32/
10. Lyashchenko A.A., Demchenko V.V., Borodavkat E.V. Geometrically Model and Computer Graphics: OpenGL Library Wikipedia: Navchalnyy posibnik. - K .: KNUBA, 2008 .-- 76 p.
11. Svirnevsky M. S., Kovalchuk C. C. Fundamentals of the development of graphical supplements: Navchalnyy posibnik - Khmelnitsky: KhNU, 2015. - 260 p.
12. Using OpenGL & GLUT in Code :: Blocks. [Electronic resource] - Access mode: http://www.sci.brooklyn.cuny.edu/~goetz/codeblocks/glut/
13. How to Make Graphics in Dev C ++ on Windows 10 [Electronic resource] - Access mode: https://www.youtube.com/watch?v=DcsAUO2QDqc
14. Reference on C / C ++. Graphic mode. [Electronic resource] - Access mode: http://mycpp.ru/cpp/scpp/cppd_graphics.h.htm
15. Borland Graphics Interface (BGI) for Windows. [Electronic resource] - Access mode: https://www.cs.colorado.edu/~main/cs1300/doc/bgi/
16. Graphics-Library. [Electronic resource] - Access mode: https://github.com/SagarGaniga/Graphics-Library
17. V.Satsyk R.Grudetsky, O.Kuzmych, N.Bahniuk, L.Hlynchuk Y.Melnychuk Reduction of Server Load by Means of CMS Drupal. IEEExplore Digital Library, Published in: 2020 10th International Conference on Advanced Computer Information Technologies, Deggendorf, 16-18 September 2020.

© Глинчук Л.Я., Гришанович Т.О., Кузьмич О.І., Багнюк Н.В