Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное общеобразовательное учреждение высшего образования «Московский государственный технический университет имени Н. Э. Баумана

(национальный исследовательский университет)»

(МГТУ им. Н. Э. Баумана)

Факультет «Робототехника и комплексная автоматизация»

Кафедра «Системы автоматизированного проектирования»

Лабораторная работа №1 по дисциплине

«Программирование графических приложений»

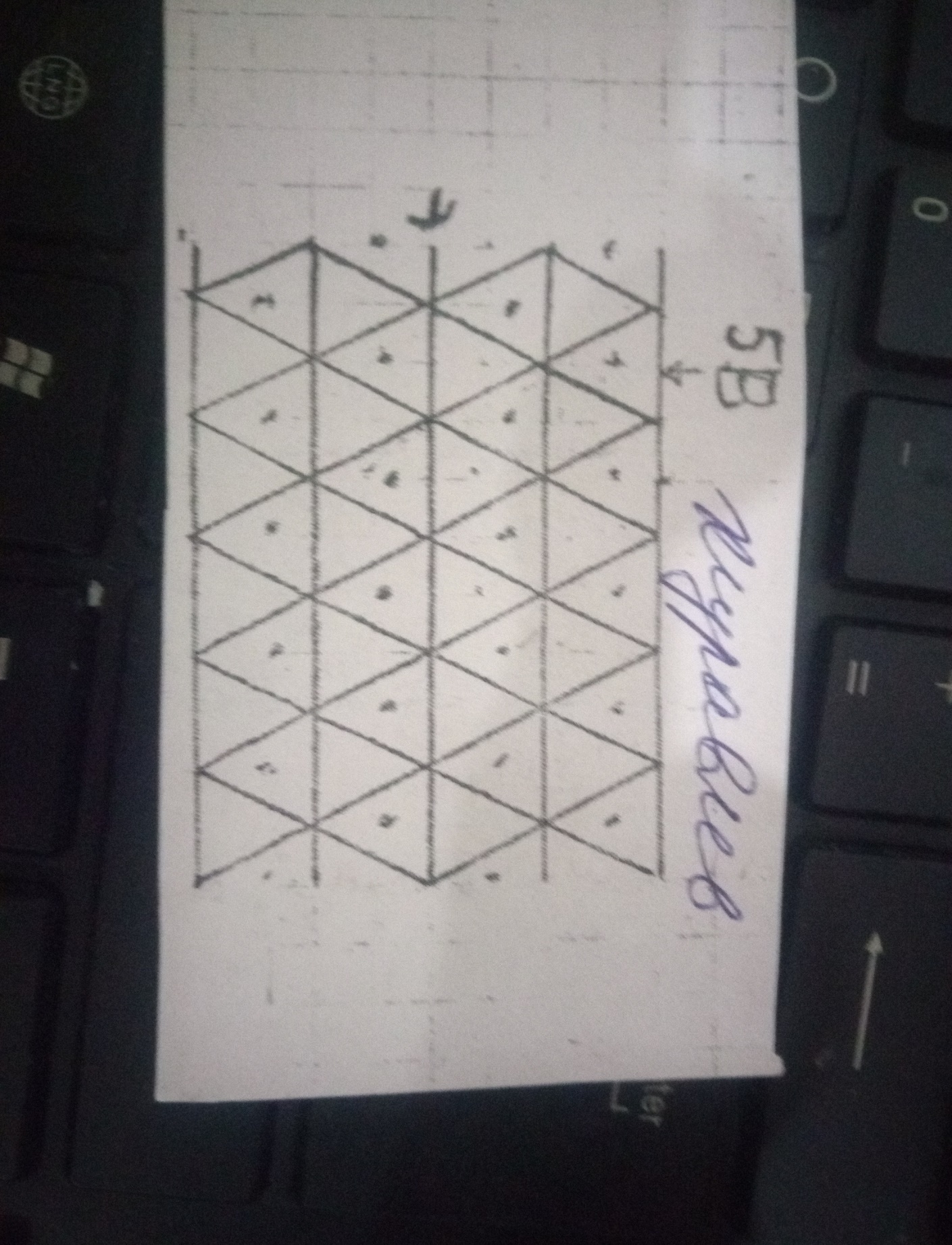
Выполнил: студент группы РК6-42Б Журавлев Н. В.

Проверил: к. т. н., доцент кафедры РК6 Волосатова Т. М.

Москва, 2021

**Текст задания**

Разработать программу для создания графического окна, фон которого отображает заданный рисунок из периодически повторяющихся геометрических фигур. Их взаимное расположение должно образовать замощение плоскости графического окна независимо от его геометрических размеров. Программную реализацию такого замощения должна обеспечивать карта пикселей минимального по размерам прямоугольного фрагмента рисунка фона, который повторяется в нем. Такая пиксельная карта должна конструироваться в программе минимальным набором примитивов графического вывода и адресоваться графическому окну структурой его атрибутов. Это должно гарантировать автоматическое тиражирование карты пикселей слево-направо и сверху-вниз в пределах графического окна, пока оно отображено на экране. Кроме того, необходимо предусмотреть возможность интерактивно инвертировать цвет рисунка фона по нажатию и отпусканию любой кнопки мыши, когда ее курсор находится в границах графического окна. Для этого в программе нужно реализовать соответствующую обработку событий и изображений для ее графического окна. При разработке программы необходимо использовать библиотечные функции базисного программного интерфейса XWindowSystem.



**Описание алгоритма**

1. Подготовить пиксельную карту рисунка повторяющегося элемента фона и присвоить ее идентификатор фоновому атрибуту графического окна.
2. Создать и визуализировать графическое окно.
3. В окне гарантируется автоматическое тиражирование заданного рисунка независимо от его исходных и текущих габаритов.

**Графическая инициализация**

1. По запросу XOpenDisplay адресуется структура дисплея.
2. Затем устанавливается по умолчанию номер экрана, его корневое окно и число цветовых плоскостей, а также графический контекст изображения.
3. Для получения этих дисплейных умолчаний вызываются соответствующие дисплейные макросы DefaultScreen, DefaultRootWindow, DefaultDepth и DefaultGC.

**Формирование пиксельной карты фона**

1. Формируется пиксельная карта повторяющегося элемента фона. При этом сначала по запросу XCreatePixmap создается пустая пиксельная карта требуемого размера (128х128) пикселей.
2. Она заполняется белым квадратом по запросам XForeground (с белым макросом WhitePixel) и XFillRectangle.
3. В ней рисуется требуемое изображение черной окружности по запросам XForeground (с черным макросом BlackPixel) и XDrawArc. В общем случае для формирования изображения могут применяться различные примитивы графического вывода, например, прямоугольники и отрезки прямых линий, которые строятся по запросам XDrawRectangle(s), XDrawLine и XDrawSegments. Вместо такой векторной техники конструирования изображения в пиксельной карте могут также использоваться растровые шаблоны из битовых карт, которые формируются, например, программой графического клиента bitmap и отображаются по запросу XCreateBitmapFromData.

**Создание графического окна**

1. По запросу XCreateWindow создается графическое окно программы с заданным начальным размером (800х600) пикселей, которое является подокном корневого окна экрана, копирует его визуальный класс и глубину.
2. Оно также декорируется оконным менеджером по его атрибуту override\_redirect со значением False. Фоновый атрибут окна background\_pixmap фиксирует идентификатор пиксельной карты из предыдущего блока. Это гарантирует автоматическое заполнение фона окна рядами изображений пиксельной карты, независимо от изменений габаритов окна, когда оно отображается на экране.
3. После создания окна, по запросу XSelectInput устанавливается маска получения событий нажатия (отпускания) кнопок мыши и клавиш на клавиатуре, когда курсор мыши находится в габаритах окна или пересекает его рамку.
4. В конце оконного блока сформированное окно отображается на экране по запросу XMapRaised, перекрывая другие окна. Графическая утилита XStoreName фиксирует заголовок окна, заданный строкой "floor", в именном свойстве WM\_NAME оконного менеджера, который отображает его в своем обрамлении окна.
5. Кроме того, по запросу XSetFunction в графическом контексте устанавливается функция инвертирования изображения GXinvert или GXcopyInverted для последующих операций графического вывода в цикле обработки событий, чтобы обеспечить переключение цвета фона графического окна.

**Обработка событий**

1. Реализует цикл чтения и обработки событий из очереди Х-сервера, которые установлены для графического окна программы в блоке 3.
2. Чтение поступающих событий обеспечивает запрос XNextEvent, а альтернативы их обработки выбираются по полю типа в структуре XEvent каждого полученного события.
3. В частности, для событий нажатия и отпускания любой кнопки мыши, когда ее курсор находится внутри окна, предусмотрена следующая обработка. Сначала инвертируется изображение в пиксельной карте фона путем копирования ее области саму на себя по запросу XCopyArea в графическом контексте с ранее установленной функцией инвертирования GXinvert (GXcopyInverted). Затем инвертированная таким образом пиксельная карта переустанавливается для фона графического окна по запросу XSetWindowBackgroundPixmap.
4. Наконец, очистка окна по запросу XClearWindow реализует инвертирование его фонового рисунка.
5. Аналогичная обработка предусматривается для события пересечения курсором рамки окна в любом направлении (EnterNotify и LeaveNotify). Итерации обработки этих мышиных событий могут продолжаться, пока значение флага завершения цикла done равно 0 (False).
6. В альтернативе KeyPress обработки нажатия любой клавиши на клавиатуре (когда окно имеет фокус ввода) этому флагу присваивается значение ее физического кода из поля keycode структуры XEvent полученного события. Его величина гарантированно больше 0 для любой клавиши. Поэтому значение флага done становится положительным (True) и происходит выход из цикла обработки событий для последующего завершения программы.

**Графический выход**

1. Предусмотрен типовой набор завершающих действий:
2. Сначала по запросу XFreePixmap освобождается память, выделенная для пиксельной карты.
3. Затем по запросу XDestroyWindow закрывается графическое окно программы.
4. Наконец, запрос XCloseDisplay обеспечивает разрыв связи с X-сервером. Физический код клавиши, которая была нажата для выхода из цикла обработки событий, возвращает exit-код программы.

**Текст программы**

#include <stdio.h>

#include <X11/Xutil.h>

#define WIDTH 74 /\* ширина повторяющегося элемента \*/

#define HEIGHT 128 /\* высота повторяющегося элемента \*/

int main(int argc, char\* argv[]) {

Display \*dpy; /\* адрес дисплейной структуры \*/

int src; /\* номер экрана дисплея \*/

GC gc; /\* графический контекст \*/

int depth; /\* число цветовых плоскостей экрана \*/

Window root; /\* корневое окно экрана \*/

Pixmap pix; /\* пиксельная карта элемента фона \*/

Window win; /\* основное окно программы \*/

XSetWindowAttributes attr; /\* атрибуты окна программы\*/

unsigned long amask; /\* маска атрибутов окна \*/

unsigned long emask; /\* маска событий \*/

XEvent event; /\* структура оконных событий \*/

unsigned int done = 0; /\* флаг завершения программы \*/

/\* Графическая инициализация \*/

dpy = XOpenDisplay(NULL); /\* Контакт с Х-сервером \*/

src = DefaultScreen(dpy); /\* номер экрана (0) \*/

depth = DefaultDepth(dpy, src); /\* глубина экрана \*/

root = DefaultRootWindow(dpy); /\* корневое окно \*/

gc = DefaultGC(dpy, src); /\* графический контекст \*/

pix = XCreatePixmap(dpy, root, WIDTH, HEIGHT, depth);

XSetBackground(dpy, gc, WhitePixel(dpy, src));

XSetForeground(dpy, gc, WhitePixel(dpy, src));

XFillRectangle(dpy, pix, gc, 0, 0, WIDTH, HEIGHT);

XSetForeground(dpy, gc, BlackPixel(dpy, src));

XDrawLine(dpy, pix, gc, WIDTH / 2, 0, WIDTH, HEIGHT / 2); /\* \ up \*/

XDrawLine(dpy, pix, gc, 0, HEIGHT / 2, WIDTH, HEIGHT / 2); /\* - middle \*/

XDrawLine(dpy, pix, gc, 0, HEIGHT / 2, WIDTH / 2, 0); /\* / up \*/

XDrawLine(dpy, pix, gc, 0, HEIGHT / 2, WIDTH / 2, HEIGHT); /\* \ down \*/

XDrawLine(dpy, pix, gc, WIDTH / 2, HEIGHT, WIDTH, HEIGHT / 2); /\* / down \*/

XDrawLine(dpy, pix, gc, 0, 0, WIDTH, 0); /\* - up \*/

XDrawPoint(dpy, pix, gc, 0, HEIGHT / 6); /\* . up \*/

XDrawPoint(dpy, pix, gc, WIDTH / 2, HEIGHT \* 9 / 13); /\* . down \*/

XFlush(dpy);

/\* или XSetFunction(dpy, gc, GXcopyInverted); \*/

amask = (CWOverrideRedirect | CWBackPixmap);

attr.override\_redirect = False; /\* Обрамление окна \*/

attr.background\_pixmap = pix; /\* Фон пиксельной карты \*/

win = XCreateWindow(dpy, root, 0, 0, 800, 600, 1, depth,

InputOutput, CopyFromParent, amask, &attr);

emask = (ButtonPressMask | ButtonReleaseMask |

EnterWindowMask | LeaveWindowMask |

KeyPressMask); /\* Маска событий \*/

XSelectInput(dpy, win, emask); /\* маскировка событий \*/

XMapRaised(dpy, win); /\* отображение окна \*/

XStoreName(dpy, win, "floor"); /\* заголовок окна \*/

XSetFunction(dpy, gc, GXinvert); /\* для инверсии фона \*/

/\* или XSetFunction(dpy, gc, GXcopyInverted); \*/

while (done == 0) {

XNextEvent(dpy, &event); /\* Чтение событий \*/

switch (event.type) {

case EnterNotify:

case LeaveNotify:

case ButtonPress:

case ButtonRelease: /\* инвертировать фон окна \*/

XCopyArea(dpy, pix, pix, gc, 0, 0,

WIDTH, HEIGHT, 0, 0);

XSetWindowBackgroundPixmap(dpy, win, pix);

XClearWindow(dpy, win);

break;

case KeyPress: /\* Выход из цикла обработки событий \*/

done = event.xkey.keycode;

break;

default:

break;

} /\* switch \*/

} /\* while \*/

XFreePixmap(dpy, pix); /\* Освободить память карты \*/

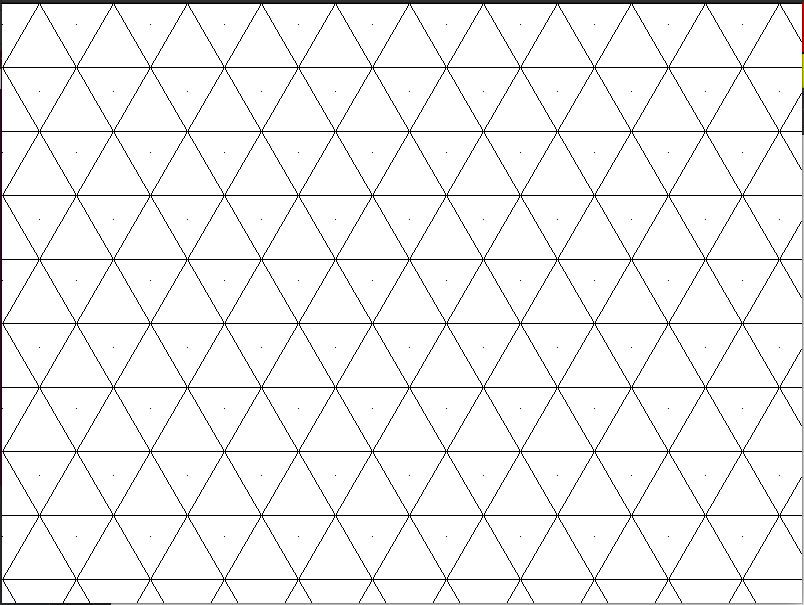
XDestroyWindow(dpy, win); /\* Закрыть окно \*/

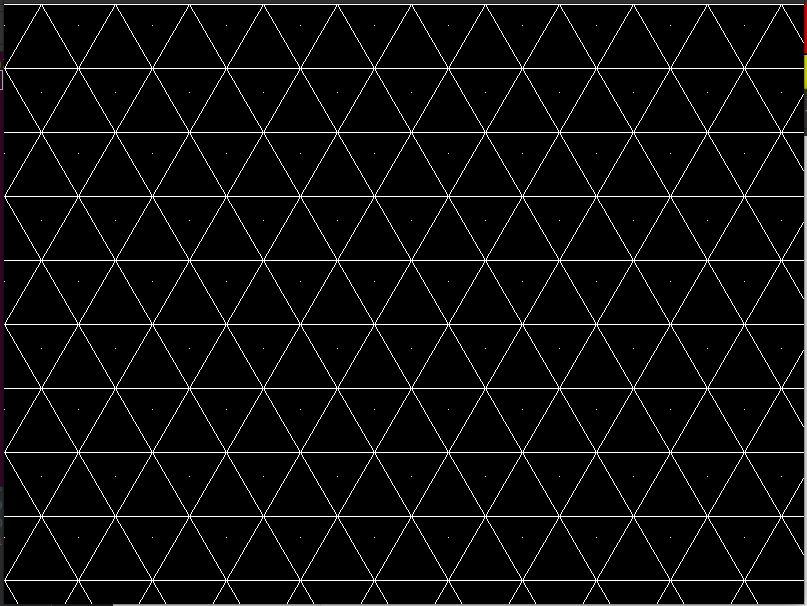
XCloseDisplay(dpy); /\* Разрыв связи с Х-сервером \*/

return done;

}

**Результат работы программы**





**Список литературы**

1. Разработка графических приложений в среде X Window System -

Д. Е. Беломойцев, Т. М. Волосатова, С. В. Родионов

1. Программирование в среде X Window на основе библиотеки Xlib - http://dfe3300.karelia.ru/koi/posob/X/index.html