

Министерство образования и науки Российской Федерации
Московский Государственный Технический Университет (МГТУ)
имени Н. Э. Баумана
Факультет «Робототехника и комплексная автоматизация (РК)»
Кафедра «Системы автоматизированного проектирования (РК6)»

**Отчет по рубежному контролю №1 по разработке графических
приложений в среде X Window System**

По курсу «Программирование графических приложений»

Вариант №W06

Выполнил:

Студент Гусаров Аркадий
Андреевич

Группа РК6-43Б

Проверил:

Дата

Подпись

Москва, 2021 г.

1. Задание:

Разработать программу редактирования 3-х цветных пиктограмм, где произвольное изображение представляется матрицей ячеек в форме изотетичных равнобедренных прямоугольников красного (Red), зеленого (Green) и синего (Blue) цвета. Размеры матрицы пиктограммы и её ячеек должны передаваться программе через аргументы командной строки. Редактирование пиктограммы должно производиться в графическом окне, которое заполнено рядами квадратных клеток без промежутков, а каждая клетка разделена диагональю на 2 треугольные ячейки. При этом диагонали в любых 2-х смежных клетках должны быть ортогональны и любые соседние треугольные ячейки являются симметрично отраженными относительно общего катета или диагонали. Сначала все ячейки должны иметь одинаковый (красный) цвет. В процессе редактирования красный, зеленый или синий цвет любой ячейки должен устанавливать щелчок по ней, соответственно, левой, средней или правой кнопки мыши. Нажатие клавиш R, G или B на клавиатуре должно устанавливать, соответственно, красный, зеленый или синий цвет для всех ячеек. Завершение программы должно обеспечивать нажатие клавиши ESC на клавиатуре. Матрица пиктограммы должна быть реализована 2-х мерным массивом квадратных окон, каждое из которых разделено диагональю на 2 все ячейки должны треугольника, закрашенных в выбранный цвет.

2. Цель работы:

Изучение базовых инструментальных средств разработки прикладного программного обеспечения в полиоконной системе X Window System. А также рассмотрение техники программирования фона графических окон с соответствующей обработкой событий и захватов указателя мыши в них библиотечными функциями программного интерфейса Xlib.

3. Алгоритм работы:

Интерфейс матрицы пиктограммы и её ячеек должны передаются программе через аргументы командной строки. Сначала все ячейки должны иметь одинаковый (красный) цвет. В процессе редактирования красный, зеленый или синий цвета любой ячейки должен устанавливать щелчок по ней, соответственно, левой, средней или правой кнопки мыши. Нажатие клавиш R, G или B на клавиатуре должно устанавливать, соответственно, красный, зеленый или синий цвет для всех ячеек. Завершение программы должно обеспечивать нажатие клавиши ESC на клавиатуре. Разработка программы ведется с применением полиоконной технологии, в соответствии с которой клетки и ряды игрового поля должны быть заданы массивами графических окон с обработкой событий и изображений в них библиотечными функциями базового программного интерфейса X Window System.

В графический модуль входят: main (основная функция), xcustom (настройка графических параметров), dispatch (диспетчер событий), repaint (функция перекраски 1 пиктограммы по нажатию мыши), redraw (изменение цвета в окне), key_analiz (анализ нажатых клавиш).

Исходный текст графического модуля начинает подключение стандартных заголовков базовой X-графики, графических утилит для взаимодействия с оконным менеджером и для макроопределения логических кодов клавиш клавиатуры.

Габариты матрицы определяются в функции main из аргумента командной строки вызова программы. Для инициализации графических матричных данных вызывается функция настройки xcustom.

4. Код программы:

- Файл pictograms.h:

```
#include <X11/Xlib.h>
#include <X11/Xutil.h>
#include <X11/keysymdef.h>
#include <stdio.h>
#include <stdlib.h>

typedef struct cell {
    Window window;
    unsigned long color;
} cell;

int xcustom();
int dispatch();
int redraw(short);
int repaint(XEvent *ev);
int key_analiz(XEvent*);
int free_custom();
int main(int, char* []);
```

- Файл pictograms.c:

```
#include "pictograms.h"

#define REDRAWALL 0
#define R 1
#define G 2
#define B 3

static Display *dpy; /* Адрес дисплейной структуры */
static GC gc; /*графический интерфейс*/
static Window desk; /* Окно игрового поля программы */
static int XO = 0; /* Начальные координаты окна */
static int YO = 0; /* программы на экране */

static unsigned ROWS; /* Число рядов клеток */
```

```
static unsigned COLS;          /* Число клеток ряда */
static unsigned SIZE;         /* Размер клетки */
static cell** box;            /* Адрес массива NYxNX клеток */
static int** tri_color;
```

```
static unsigned long color[5];
```

```
XPoint *points1;
XPoint *points2;
XPoint *points3;
XPoint *points4;
```

```
/* Настройка графических параметров */
```

```
int xcustom() {
    int x, y; /* Позиции окон */
    unsigned w, h; /* Габариты окон */
    int depth = DefaultDepth(dpy, 0); /* Глубина экрана 0 */
    Window root; /* Корневое окно экрана */
    XSetWindowAttributes attr; /* Атрибуты окон */
    unsigned long amask; /* Маска оконных атрибутов */
    XSizeHints hint; /* Геометрия оконного менеджмента */
    int i, j; /* индексы окон */
    Colormap colormap;
```

```
/* Настройка графических контекстов */
```

```
    root = DefaultRootWindow(dpy); /* Корневое окно экрана */
    gc = XCreateGC(dpy, root, 0, NULL);
```

```
/*настройка цветов*/
```

```
    colormap = XDefaultColormap(dpy, DefaultScreen(dpy));
    XColor RGBpixel;
    XParseColor(dpy, colormap, "#ffffff", &RGBpixel);    //white
    XAllocColor(dpy, colormap, &RGBpixel);
    color[0] = RGBpixel.pixel;
    XParseColor(dpy, colormap, "#ff0000", &RGBpixel);    //red
    XAllocColor(dpy, colormap, &RGBpixel);
    color[1] = RGBpixel.pixel;
    XParseColor(dpy, colormap, "#00ff00", &RGBpixel);    //green
    XAllocColor(dpy, colormap, &RGBpixel);
    color[2] = RGBpixel.pixel;
    XParseColor(dpy, colormap, "#0000ff", &RGBpixel);    //blue
    XAllocColor(dpy, colormap, &RGBpixel);
```

```

color[3] = RGBpixel.pixel;
XParseColor(dpy, colormap, "#000000", &RGBpixel);    //black
XAllocColor(dpy, colormap, &RGBpixel);
color[4] = RGBpixel.pixel;
/*настройка цветов*/

/* Настройка игрового окна программы */
attr.override_redirect = False; /* WM обрамление окна */
attr.background_pixel = color[0]; //0xFFFFFF; /* white */
amask = (CWOVERRIDE_REDIRECT | CWBACKPIXEL);
w = COLS * SIZE; /* Габариты */
h = ROWS * SIZE; /* игрового окна */
x = X0; y = Y0; /* Начальные координаты окна игры */
desk = XCreateWindow(dpy, root, x, y, w, h, 1, depth, InputOutput,
                    CopyFromParent, amask, &attr); /* Геометрические рекомендации оконного
менеджера */
hint.flags = (PMinSize | PMaxSize | PPosition);
hint.min_width = hint.max_width = w; /* ФИКСИРОВАТЬ */
hint.min_height = hint.max_height = h; /* габариты и */
hint.x = x; hint.y = y; /* позицию окна игрового поля */
XSetNormalHints(dpy, desk, &hint); /* в свойстве WM */
XStoreName(dpy, desk, "Pictograms Editor"); /* Заголовок окна */

/* Настройка окон клеток */
int x1, y1, x2, y2;
amask = CWOVERRIDE_REDIRECT | CWBACKPIXEL | CWEVENTMASK;
attr.override_redirect = True; /* Отмена обрамления окна */
attr.background_pixel = color[4];
attr.event_mask = (KeyPressMask | ExposureMask | ButtonPressMask);
w = SIZE;
h = SIZE; /* Габариты окна клетки */
x = 0;
y = 0;
box = (cell **)calloc(ROWS, sizeof(cell*));
tri_color = (int **)calloc(ROWS * COLS, sizeof(int*));
for(i = 0; i < ROWS; i++) { /* Цикл по рядам клеток */
    box[i] = (cell *) calloc(COLS, sizeof(cell));
    x = 0;
    for (j = 0; j < COLS; j++) { /* Создать окна клеток */
        box[i][j].window = XCreateWindow(dpy, desk, x, y, w, h, 0, depth, InputOutput, CopyFromParent,
amask, &attr); /* Отображение всех окон на экране */
        box[i][j].color = 0;
    }
}

```

```

        tri_color[i * COLS + j] = (int *) calloc(2, sizeof(int));
        tri_color[i * COLS + j][0] = 1;
        tri_color[i * COLS + j][1] = 1;
        x += SIZE;
    } /* for */
    y += SIZE;
} /* for */

/*создание окон*/
XMapWindow(dpy, desk);
XMapSubwindows(dpy, desk);
return(0);
}

/* xcustom */

int dispatch() { /* Диспетчер событий */
    XEvent event; /* Структура событий */
    int done = 0; /* Флаг выхода */
    while(done == 0) { /* Цикл обработки событий */
        XNextEvent(dpy, &event); /* Чтение событий */
        switch(event.type) {
            case Expose:
                redraw(R); /* Начальная перерисовка */
                break;
            case ButtonPress:
                repaint(&event); /*перекраска региона*/
                break;
            case KeyPress:
                done = key_analiz(&event); /*проверка условий нажатий на клавиши*/
                break;
            default: break;
        } /* switch */
    } /* while */
    return(0);
} /* dispatch */

/*функция перекраски 1 пиктограммы по нажатию мыши*/
int repaint(XEvent* ev) {
    int clr;

    for (int i = 0; i < ROWS; i++)
        for (int j = 0; j < COLS; j++)
            {

```

```

if (box[i][j].window == ev->xbutton.window)
{
    if (i % 2 == 0) {
        if (j % 2 == 0) {
            if (ev->xbutton.x + ev->xbutton.y <= SIZE) {
                clr = ((tri_color[i * COLS + j][0] + 1) > 3) ? 1 : tri_color[i * COLS +
j][0] + 1;

                tri_color[i * COLS + j][0] = clr;
                XSetForeground(dpy, gc, color[clr]);
                XFillPolygon(dpy, box[i][j].window, gc, points1, 3, Convex,
CoordModeOrigin);
            } else {
                clr = ((tri_color[i * COLS + j][1] + 1) > 3) ? 1 : tri_color[i * COLS +
j][1] + 1;

                tri_color[i * COLS + j][1] = clr;
                XSetForeground(dpy, gc, color[clr]);
                XFillPolygon(dpy, box[i][j].window, gc, points2, 3, Convex,
CoordModeOrigin);
            }
        } else {
            if (ev->xbutton.y >= ev->xbutton.x) {
                clr = ((tri_color[i * COLS + j][0] + 1) > 3) ? 1 : tri_color[i * COLS +
j][0] + 1;

                tri_color[i * COLS + j][0] = clr;
                XSetForeground(dpy, gc, color[clr]);
                XFillPolygon(dpy, box[i][j].window, gc, points4, 3, Convex,
CoordModeOrigin);
            } else {
                clr = ((tri_color[i * COLS + j][1] + 1) > 3) ? 1 : tri_color[i * COLS +
j][1] + 1;

                tri_color[i * COLS + j][1] = clr;
                XSetForeground(dpy, gc, color[clr]);
                XFillPolygon(dpy, box[i][j].window, gc, points3, 3, Convex,
CoordModeOrigin);
            }
        }
    } else {
        if (j % 2 == 0) {
            if (ev->xbutton.y >= ev->xbutton.x) {
                clr = ((tri_color[i * COLS + j][0] + 1) > 3) ? 1 : tri_color[i * COLS +
j][0] + 1;

                tri_color[i * COLS + j][0] = clr;
                XSetForeground(dpy, gc, color[clr]);

```



```

        tri_color[i * COLS + j][1] = type;
        XSetForeground(dpy, gc, color[type]);
        if (i % 2 == 0) {
            if (j % 2 == 0) {
                XFillPolygon(dpy, box[i][j].window, gc, points1, 3, Convex,
CoordModeOrigin);

                XFillPolygon(dpy, box[i][j].window, gc, points2, 3, Convex,
CoordModeOrigin);

            } else {
                XFillPolygon(dpy, box[i][j].window, gc, points3, 3, Convex,
CoordModeOrigin);

                XFillPolygon(dpy, box[i][j].window, gc, points4, 3, Convex,
CoordModeOrigin);

            }
        } else {
            if (j % 2 == 0) {
                XFillPolygon(dpy, box[i][j].window, gc, points3, 3, Convex,
CoordModeOrigin);

                XFillPolygon(dpy, box[i][j].window, gc, points4, 3, Convex,
CoordModeOrigin);

            } else {
                XFillPolygon(dpy, box[i][j].window, gc, points1, 3, Convex,
CoordModeOrigin);

                XFillPolygon(dpy, box[i][j].window, gc, points2, 3, Convex,
CoordModeOrigin);

            }
        }
    }
    break;
}
}

return 0;
}

```

/*обработка событий нажатия на клавиши*/

```

int key_analiz(XEvent* ev) {
    if (ev->xkey.keycode == XKeysymToKeycode(dpy, XK_Escape))
        return 1;

    if ((ev->xkey.keycode == XKeysymToKeycode(dpy, XK_r)) || (ev->xkey.keycode ==
XKeysymToKeycode(dpy, XK_R))) {
        redraw(R);
    }
}

```

```

        return 0;
    }

    if ((ev->xkey.keycode == XKeysymToKeycode(dpy, XK_g)) || (ev->xkey.keycode ==
XKeysymToKeycode(dpy, XK_G))) {
        redraw(G);
        return 0;
    }

    if ((ev->xkey.keycode == XKeysymToKeycode(dpy, XK_b)) || (ev->xkey.keycode ==
XKeysymToKeycode(dpy, XK_B))) {
        redraw(B);
        return 0;
    }
    return 0;
}

/*освобождение памяти выделенной для массивов окон*/
int free_custom() {
    for ( int i = 0; i < ROWS; i++)
        free(box[i]);
    free(box);
    return 0;
}

int main(int argc, char* argv[]) {
    if(argc != 4) {
        fprintf(stderr, "Default: ./pics 100 10 10\n");
        SIZE = 100;
        ROWS = 10;
        COLS = 10;
    } else {
        SIZE = atoi(argv[1]); //дефолтные значения
        ROWS = atoi(argv[2]); //дефолтные значения
        COLS = atoi(argv[3]); //дефолтные значения
    }

    XPoint p1[] = { { SIZE - 2, 1 }, { 1, 1 }, { 1, SIZE - 2 } };
    XPoint p2[] = { { SIZE - 1, 2 }, { SIZE - 1, SIZE - 1 }, { 2, SIZE - 1 } };
    XPoint p3[] = { { 2, 1 }, { SIZE - 1, 1 }, { SIZE - 1, SIZE - 2 } };
    XPoint p4[] = { { 1, 2 }, { 1, SIZE - 1 }, { SIZE - 2, SIZE - 1 } };

    points1 = p1;

```

```

        points2 = p2;
        points3 = p3;
        points4 = p4;

        dpy = XOpenDisplay(NULL);
        xcustom();
        dispatch();
        XDestroySubwindows(dpy, desk);
        XDestroyWindow(dpy, desk);
        free_custom();
        XCloseDisplay(dpy);
        return(0);
    } /* main */

```

5. Список литературы:

1. O'Reilly & Associates, Inc. - Table of contents for Xlib Programming Manual (O'Reilly & Associates, Inc.): Режим доступа к ст. http://www.sbin.org/doc/Xlib/index_contents.html.
2. Adrian Nye - Volume One: Xlib Programming Manual: Режим доступа к ст. http://www.ac3.edu.au/SGI_Developer/books/XLib_PG/sgi_html.
3. Вадим Годунко - Xlib - интерфейс с X Window на языке C : Режим доступа к ст. <http://motif.opennet.ru/book3.html>.
4. Kluwer Academic Publishers - Fundamentals of X Programming GUI and Beyond.pdf Режим доступа <http://ftp.homei.net.ua/index>.
5. Kenton Lee - Technical Window System and Motif WWW Sites: Режим доступа к ст. <http://www.rahul.net/kenton/xsites.html>.
6. Robert W. Scheifler - RFC 1013 - X Window System Protocol. Режим доступа к ст. <http://www.apps.ietf.org/rfc/rfc1013.html>.
7. Theo Pavlidis - Fundamentals of X Programming GUI and Beyond. Режим доступа <http://www.maives.ru/modules/news>.