

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 №2 по разработке графических  
приложений в среде X Window System**

По курсу «Программирование графических приложений»

Выполнил:

Студент Гусаров Аркадий  
Андреевич

Группа РК6-43Б

Проверил:

Дата

Подпись

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Москва, 2021 г.

## **1. Задание:**

Разработать программу, которая отображает процесс распространения поперечных волн в форме концентрических кругов, синхронно расходящихся от точечного источника, произвольно расположенного в плоскости графического окна фиксированного размера. Источник волны может быть установлен в любой точке графического окна, которую указывает курсор мыши при нажатии на ней левой кнопки. При этом курсор мыши меняет форму и фиксируется в точке источника волны, пока левая кнопка мыши удерживается в нажатом состоянии. Кроме того, необходимо обеспечить периодические колебания курсора в окрестности точки источника волны с любой визуально различимой частотой и амплитудой. При каждом таком колебании курсора мыши образуется волновой круг, который распространяется в плоскости графического окна с постоянной фазовой скоростью, пока полностью не исчезает за его границами. Образование волновых кругов и колебания курсора мыши должны прекратиться, когда нажатая левая кнопка мыши будет освобождена. В этот момент курсор мыши освобождается от захвата и возвращает исходную форму, а волновой процесс постепенно затухает. Он окончательно завершается, когда все его круги исчезают за границами графического окна программы. При этом не должно быть никаких препятствий начинать новые волновые процессы до окончательного затухания любого из предыдущих. Завершение программы должно обеспечивать нажатие комбинации клавиш CTRL+ALT+F на клавиатуре. При разработке программы необходимо предусмотреть соответствующую обработку событий и изображений в её графическом окне, используя библиотечные функции программного интерфейса Xlib из состава X Window System.

## **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:**

В программе используется функция рисования в контексте главного окна, выполняющая отрисовку основываясь на массиве данных, которые могут изменяться в ходе работы программы.

Далее будут кратко описаны функции, которые используются в программе, в том порядке, в котором они идут.

Сначала получаем предварительные сведения из дефолтных настроек.

XOpenDisplay - установление связи с графическим терминалом (X-сервером).

DefaultRootWindow - макрос возвращает корневое окно для экрана по умолчанию.

DefaultScreen - макрос возвращает номер экрана в подпрограмме XopenDisplay.

DefaultDepth - макрос возвращает глубину цвета (количество разрядов, относящихся для кодирования цвета в окне) для указанного экрана.

DefaultVisual - макрос возвращает визуальные характеристики окна для указанного экрана.

Создаем графический образ.

XCreatePixmap - создает графический образ с указанными размерами и глубиной цветности, возвращая его идентификатор.

Имеет 5 аргументов:

- display - указатель на структуру, описывающую соединение с X-сервером;
- d - окно или графический образ для определения экрана сервера;
- width and height - размеры графического образа;
- depth - глубина цветности графического образа.

DefaultGC - возвращает графический контекст по умолчанию для корневого окна указанного экрана.

Сначала выполняется инициализация массива данных, затем при вызове события перерисовки функция отрисовки выполняет рисование треугольных элементов учитывая их смещение и положение диагонали в клетке между треугольниками.

XSetBackground(display, gc, background) - устанавливает фоновый цвет для графического контекста gc.

Background - индекс фонового цвета.

XSetForeground(display, gc, foreground) - устанавливает цвет графических примитив при выводе их через графический контекст gc.

Foreground - индекс цвета для вывода графических примитивов

XFillRectangle(display, d, gc, x, y, width, height) - рисует заполненный прямоугольник, левый верхний угол которого находится в координатах (x, y), а размеры равны (width, height).

XDrawLine(display, d, gc, x1, y1, x2, y2) - рисует прямую линию между двумя указанными точками (x1, y1), (x2, y2).

XDrawPoint(display, d, gc, x, y) - рисует точку в указанных координатах. Так как точка на графическом окне отображается очень мелко - была использована функция рисования дуги.

XFillArc(display, d, gc, x, y, width, height, angle1, angle2) - рисует дугу, образованную бесконечно тонким эллипсом, вписанным в прямоугольник.

Заполняем атрибуты окна.

С помощью маски меняем некоторые параметры из аргумента attributes, а именно:

- override\_redirect - размер окна и его положение смогут меняться только с помощью менеджера окон, так как мы присваиваем ему значение False.
- background\_pixmap - фон окна будет задаваться картинкой (картой пикселей).

Создаем графическое окно.

XCreateWindow - создает включенное окно-потомок для указанного окна-предка, возвращая идентификатор созданного окна.

Имеет 8 аргументов:

- display - указатель на структуру, описывающую соединение с X-сервером;
- parent - идентификатор родительского окна;
- x, y - координата левого верхнего угла;
- width and height - размеры графического образа;
- border\_width - толщина рамки вокруг окна;
- depth - количество разрядов, отводящихся для кодирования цвета в окне;
- class - класс окна. Допустимые значения InputOutput, InputOnly,

CopyFromParent;

- visual - визуальные характеристики окна;

- valuemask - маска, указывающая, какие параметры из аргумента attributes следует указывать;
- attributes - расширенные параметры окна.

XStoreName - устанавливает заголовок окна.

XMapWindow - делает указанное окно видимым.

XFlush - принудительная передача содержимого буфера вывода.

Потом прописываем маску событий и функцию инверсии фона.

Затем следует цикл while, реагирующий на изменение положение курсора в поле графического окна и за его пределами. При нажатии любой клавиши на клавиатуре происходит выход из графического окна.

XDestroyWindow - удаляет окно.

XCloseDisplay - прерывает соединение с X-сервером.

#### 4. Код программы:

- Файл 23M.h:

```
#define DELAY (1 << 14)

/*структура стрелы*/
typedef struct
{
    int x;    //координаты
    int y;    //центра
    int R;    //текуий радиус
    int dR; //частота
    int w;    //фазовая скорость
    int n;
    int m;
} XArrow;

/* Spiral Implementation function */

int configure_arrow(XEvent *, XArrow *, int, int);           /* задание точек стрелки */
int redraw(XEvent *, GC, GC, XArrow *, int, int);          /* перерисовка потерянных
фрагментов окна */
int draw_arrow(Display *, Window, GC, GC, XArrow *, int, int); /*отрисовка стрелы в окне*/
int key_check(XEvent *);
/* ответ на события нажатия клавиш */
int maxi(int, int, int, int);
int side(int, int, int, int);
```

- Файл 23M1.c:

```
/* 2 centred spiral implementation */

#include <X11/Xlib.h>
```

```

#include <X11/keysym.h>
#include <X11/keysymdef.h>
#include "23M.h"
#include <math.h>

int side(int x1, int x2, int y1, int y2)
{
    return ((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2));
}

int maxi(int x, int y, int w, int h)
{
    int m;

    if (side(x, 0, y, 0) > side(x, 0, y, h))
    {
        m = side(x, 0, y, 0);
    }
    else
    {
        m = side(x, 0, y, h);
    }
    if (m < side(x, w, y, 0))
    {
        m = side(x, w, y, 0);
    }
    if (m < side(x, w, y, h))
    {
        m = side(x, w, y, h);
    }
    return sqrt(m);
}

/* задание точек стрелки */
int configure_arrow(XEvent *ev, XArrow *pr, int w, int h)
{
    pr->x = ev->xbutton.x;
    pr->y = ev->xbutton.y;
    pr->R = 0;
    pr->dR = 20;
    pr->w = 10;
    pr->n = 1;
    pr->m = maxi(pr->x, pr->y, w, h) / pr->dR;
    return 0;
}

```

```

/*отрисовка стрелы в окне*/
int draw_arrow(Display *dpy, Window win, GC Bgc, GC Fgc, XArrow *pr, int w, int h)
{

    int n;
    int r = pr->dR;
    for (int i = 0; i < pr->n; i++)
    {
        XDrawArc(dpy, win, Bgc, pr->x + 10 * i, pr->y + 10 * i, pr->R - r * i, pr->R - r * i, 0 * 64, 360 *
64);

    }
    pr->R += pr->w;
    n = (int)pr->R / pr->dR;
    n++;
    if (2 * (pr->m + 3) >= n)
    {
        pr->n = n;
    }
    pr->x -= 5;
    pr->y -= 5;
    for (int j = 0; j < pr->n; j++)
    {
        XDrawArc(dpy, win, Fgc, pr->x + 10 * j, pr->y + 10 * j, pr->R - r * j, pr->R - r * j, 0 * 64, 360 *
64);

    }
    XFlush(dpy);
}

/* перерисовка потерянных фрагментов окна */
int redraw(XEvent *ev, GC Bgc, GC Fgc, XArrow *pr, int w, int h)
{
    static XRectangle clip[32]; /* clip rectangles buffer */
    static int n = 0;          /* clip rectangles number */
    clip[n].x = ev->xexpose.x;
    clip[n].y = ev->xexpose.y;
    clip[n].width = ev->xexpose.width;
    clip[n].height = ev->xexpose.height;
    n++;
    if ((ev->xexpose.count > 0) && (n < 32))
        return (0);
    XSetClipRectangles(ev->xexpose.display, Fgc, 0, 0, clip, n, Unsorted);

    XGCValues gv;

```

```

        gv.line_style = LineOnOffDash;
        XChangeGC(ev->xexpose.display, Fgc, GCLineStyle, &gv);
        gv.line_style = LineSolid;
        XChangeGC(ev->xexpose.display, Fgc, GCLineStyle, &gv);
        draw_arrow(ev->xexpose.display, ev->xexpose.window, Bgc, Fgc, pr, w, h);

        XSetClipMask(ev->xexpose.display, Fgc, None);
        return (n = 0);
    } /* redraw */

    /* ответ на события нажатия клавиш */
    int key_check(XEvent *ev)
    {
        KeySym ks = XLookupKeysym((XKeyEvent *)ev, 0);
        // if ((ks == XK_Alt_L) && (ev->xkey.state == ControlMask))
        if (ks == XK_Alt_L)
        {
            return 0; /*код выхода*/
        }
        return -1;
    } /* rapid */

```

- **Файл 23M2.c:**

```

    /* 2 centred spiral main function */

#include <X11/Xlib.h>
#include <X11/Xutil.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "23M.h"

/* Main function */

int main()
{
    Display *dpy; /* Graphic Display */
    Window win; /* programm main window */
    GC gc[2]; /* Black & White Graphic Context */
    int scr; /* screen number = 0 */
    XArrow r; /* spiral structure */
    XArrow *frag = (XArrow *)malloc(sizeof(XArrow));
    XArrow *p;
    int nfrag = 0;

```

```

{          /* Display Block */
    unsigned long tone; /* Light drawing tone */
    dpy = XOpenDisplay(NULL);
    scr = DefaultScreen(dpy);
    win = DefaultRootWindow(dpy);
    scr = DefaultScreen(dpy);
    gc[0] = XCreateGC(dpy, win, 0, 0);
    gc[1] = XCreateGC(dpy, win, 0, 0);
    tone = WhitePixel(dpy, scr); /* = 0xFFFFFF; */
    XSetForeground(dpy, gc[0], tone);

} /* Display block */

{          /* Window block */
    unsigned w = 640, h = 480; /* main window width & height */
    XSetWindowAttributes attr; /* window attributes structure */
    XGCValues gval;          /* GC structure */
    unsigned long amask;      /* window attributes mask */
    Window root = win;        /* Display root window */
    XSizeHints hint;          /* Geometry WM hints */

    /*создание окна*/
    amask = (CWOVERRIDE_REDIRECT | CWBACKPIXEL);
    XGetGCValues(dpy, gc[1], GC_BACKGROUND, &gval);
    attr.background_pixel = gval.background; /* = 0x0 */
    attr.override_redirect = False;
    win = XCreateWindow(dpy, root, 0, 0, w, h, 1, CopyFromParent,
                        InputOutput, CopyFromParent, amask, &attr);

    /*задание фиксированного размера окна*/
    hint.flags = (PMIN_SIZE | PMAX_SIZE);
    hint.min_width = hint.max_width = w;
    hint.min_height = hint.max_height = h;
    XSetNormalHints(dpy, win, &hint);
    XStoreName(dpy, win, "spiral");
    XMapWindow(dpy, win);

} /* window block */

{ /* Multi Block */

    unsigned long emask; /* window event mask */
    XEvent event;        /* graphic event structure */

```



```

int freeze = -1;    /* window visibility stop state */
unsigned delay = DELAY; /* multi delay period = 2^rate */
int multi = (1 << 12); /* multi code */
int count = 0;      /* delay count */
int w = 500, h = 500;
int reconf;

emask = (EnterWindowMask | LeaveWindowMask | ButtonPressMask | ButtonReleaseMask |
ExposureMask | StructureNotifyMask | KeyPressMask);
XSelectInput(dpy, win, emask);

while (multi != 0)
{ /* Async dispatch event with multic ground */
    event.type = 0;
    XCheckWindowEvent(dpy, win, emask, &event);
    switch (event.type)
    {
    case Expose:
        redraw(&event, gc[1], gc[0], &r, w, h);
        break;
    case ButtonPress:
        nfrag++;
        p = (XArrow *)malloc(sizeof(XArrow));
        configure_arrow(&event, p, w, h);
        frag = realloc(frag, (nfrag + 1) * sizeof(XArrow));
        memcpy((frag + nfrag - 1), p, sizeof(XArrow));
        break;
    case LeaveNotify:
        freeze = -1;
        break;
    case EnterNotify:
        freeze = 1;
    case KeyPress:
        multi = key_check(&event);
        break;
    default:
        break;
    } /* switck /
    }*/

    if (freeze < 0) /* Freeze display spiral */
    {
        continue;
    }
}

```

```

        if (count++ < delay) /* Delay display spiral */
        {
            continue;
        }
        count = 0; /* reset count to next delay */
        if (freeze == 1)
        {
            for (int i = 0; i < nfrag; i++)
            {
                draw_arrow(dpy, win, gc[1], gc[0], &frag[i], w, h); //отрисовка треугольника
            }
        }
    } /* while event */
} /* multi block */

{ /* Exit block */
    XDestroyWindow(dpy, win);
    XCloseDisplay(dpy);
    free(frag);
    return (0);
} /* exit block */

} /* 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](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](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>.