



ปีการศึกษา 2533
ปริญาานิพนธ์เรื่อง

WINDOW INTERFACE





WINDOW INTERFACE

(source code)

โดย

นาย พิริยะ เกวลี 30.1190

นาย พิรัตน์ ศิลประชาวงค์ 30.1191

อาจารย์ที่ปรึกษา

ดร. บุญฤทธิ์ เครือตราฐ

ปริญญาโท สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

ปีการศึกษา 2533

เลขที่	T33013 พ 4
เลขทะเบียน	027846
วัน, เดือน, ปี	12 ก.ค. 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำเป็เซ

027846

ปริญญานิพนธ์ปีการศึกษา 2533

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง WINDOW INTERFACE (source code)

ผู้จัดทำ

1. นาย พิริยะ เกลือ 30.1190

2. นาย นิตินันต์ ศิลประชาวงษ์ 30.1191

อาจารย์ที่ปรึกษา

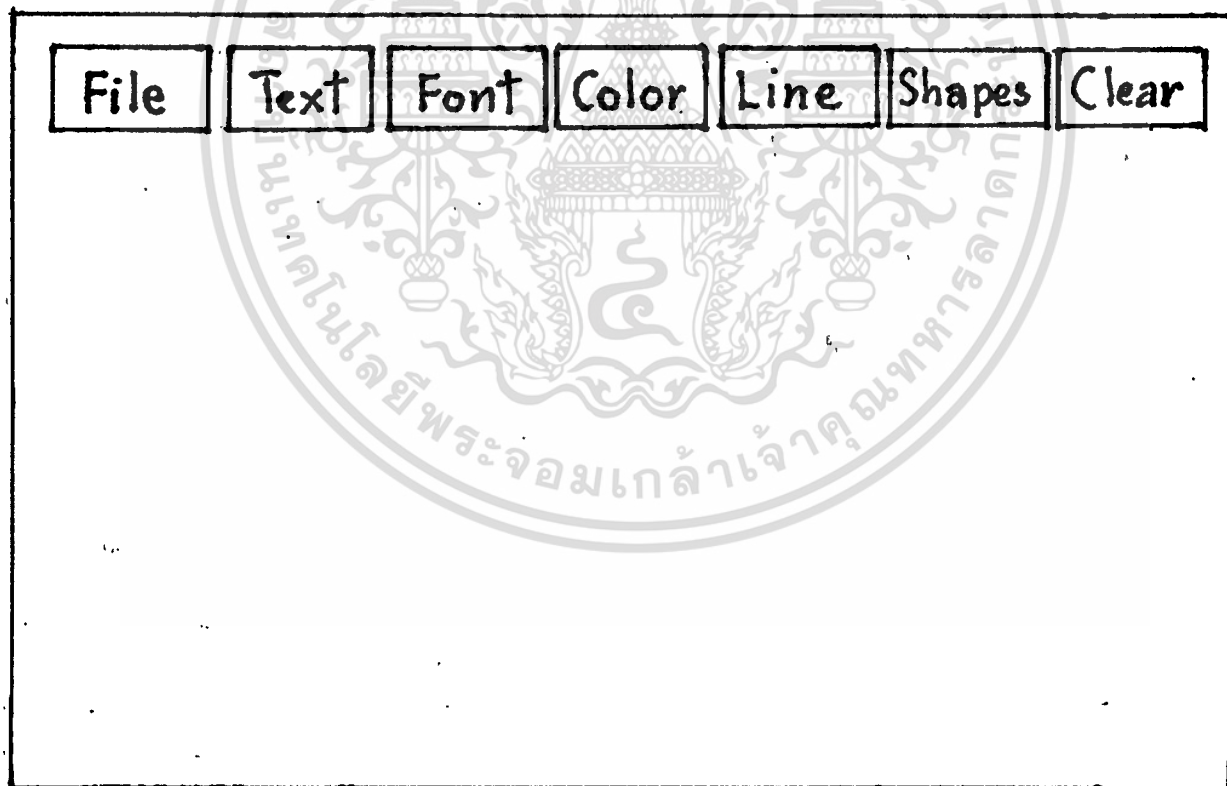
(ดร. บุญฤทธิ์ เครือตราชู)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งานโปรแกรมกราฟฟิคเอดิเตอร์

สำหรับฟังก์ชันการทำงานหลักจะประกอบด้วย 7 ฟังก์ชัน คือ

1. ฟังก์ชันเกี่ยวกับไฟล์
 2. ฟังก์ชันการเขียนตัวอักษร
 3. ฟังก์ชันเลือกแบบตัวอักษร
 4. ฟังก์ชันการชี้ทสี
 5. ฟังก์ชันการเลือกลักษณะและขนาดเส้น
 6. ฟังก์ชันการวาดรูป
 7. ฟังก์ชันการเคลียร์ (clear) หน้าจอ
- ซึ่งมีลักษณะเมนูดังรูปที่ 1



รูปที่ 1 แสดงเมนูหลักของโปรแกรมกราฟฟิคเอดิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันเกี่ยวกับไฟล์

เมื่อใช้เมาส์คลิก (click) ที่เมนูไฟล์ (File) จะปรากฏหน้าต่างเมนูขึ้น ซึ่งประกอบด้วย

1. การโหลดไฟล์ (Load)
2. การเซฟไฟล์ (Save)
3. การไดเรกทอรีไฟล์ (Dir)
4. การออกจากโปรแกรม (Quit)

ซึ่งในส่วนที่ใช้งานได้มีเพียงข้อ 4 เท่านั้น เนื่องจากเวลาไม่เพียงพอที่จะทำให้เสร็จสมบูรณ์

ฟังก์ชันการเขียนตัวอักษร

เมื่อใช้เมาส์คลิกที่เมนูเท็กซ์ (Text) นี้ สัญลักษณ์ของเคอร์เซอร์จะเปลี่ยนจากเครื่องหมายบวกเป็นเครื่องหมายลักษณะคล้ายตัวอักษรโอใหญ่ ซึ่งสามารถพิมพ์ข้อความให้ปรากฏได้ตามตำแหน่งของเคอร์เซอร์ขณะนั้น เมื่อกดคีย์เอ็นเตอร์ (enter key) ข้อความจะถูกเก็บบันทึกไว้และเคอร์เซอร์จะขึ้นบรรทัดใหม่ให้

ฟังก์ชันการเลือกแบบตัวอักษร

เมื่อคลิกที่เมนูฟอนต์ (Font) จะปรากฏเมนูขึ้น ประกอบด้วยลักษณะตัวอักษร 4 แบบ

1.8x13

2.8x13bold

3.9x15

4.thai24

แบบที่ 1 ถึง 3 เป็นตัวอักษรภาษาอังกฤษ และแบบที่ 4 เป็นภาษาไทย ซึ่งยังไม่สมบูรณ์เท่าที่ควรแต่ก็สามารถใช้ได้ในระดับหนึ่ง

ฟังก์ชันการเช็ทสี

เมื่อคลิกที่เมนูคาลเลอร์ (Color) จะปรากฏเมนู ประกอบด้วยสีต่างๆ 5 สี

1. สีแดง (Red)
2. สีเขียว (Green)
3. สีฟ้า (Blue)
4. สีเหลือง (Yellow)
5. สีดำ (Black)

เมื่อใช้เมาส์คลิกที่พาดำเนินเมนูใด สีโฟร์กราวนด์ (foreground) ที่ปรากฏจะเป็นสีดังกล่าวตามที่ต้องการ

ฟังก์ชันการเลือกลักษณะและขนาดเส้น

เมื่อคลิกเมาส์ที่เมนูไลน์ (Line) จะปรากฏเมนู คือ

1. Style
2. Width

ซึ่งมีเมนูย่อยลงไปอีก สำหรับเมนูสไตล์ (Style) จะมีลักษณะเส้น 3 แบบ คือ

1. Solid
2. Dash1
3. Dash2

ส่วนเมนูวิธต์ (Width) ประกอบด้วยขนาดเส้น 3 ขนาด คือ 1, 2 และ 3 ซึ่งถ้าใช้ขนาดเส้น 2 และ 3 การทำงานของโปรแกรมจะช้าลงอย่างเห็นได้ชัด เนื่องจากเครื่องจะทำงานไม่ทันกับการเคลื่อนที่ของพอยน์เตอร์เมื่อวาดรูป ตำแหน่งที่รูปปรากฏอาจจะผิดพลาดไปบ้างเล็กน้อย

ฟังก์ชันการวาดรูป

เมื่อคลิกเมาส์ที่เมนูเชปส์ (Shapes) จะปรากฏพูลดาวน์เมนูประกอบด้วย

- 1.Point
- 2.Line
- 3.Circle
- 4.N-tangle
- 5.Rectangle

ในการวาดจุดและเส้นก็เพียงแต่กำหนดจุดให้เท่านั้นโดยใช้เมาส์ปุ่มที่ 1 การวาดรูปหลายเหลี่ยมจะใช้เมาส์ปุ่มที่ 3 เป็นตัวบอกการสิ้นสุด โดยจะลากเส้นตรงระหว่างจุดสุดท้ายที่กำหนดและจุดแรกที่กำหนด ส่วนการวาดวงกลมและสี่เหลี่ยมจะมีเส้นอ้างอิงบอกขนาดในขณะที่พอยน์เตอร์เคลื่อนที่เสมอ

เมื่อมีการคลิกเมนู เมนูที่ถูกคลิกจะถูกอินเวอร์ท (inverted) ระหว่างไฟร์กราวนด์ และแบ็คกราวนด์ (background) สำหรับเมนูหลักที่มีพูลดาวน์เมนูปรากฏขึ้นเมื่อเลือกการทำงานตามต้องการแล้วจะต้องคลิกเมาส์อีก 1 ที เพื่ออันแมป (unmap) เมนูเหล่านั้นจากหน้าจอ จากนั้นจึงจะทำงานตามปกติได้คือ การวาดรูปหรือการเขียนตัวอักษร

ฟังก์ชันการเคลียร์หน้าจอ

เมื่อคลิกเมนูเคลียร์ (Clear) จะเป็นการลบสิ่งที่ปรากฏบนหน้าจอทั้งหมดทั้งตัวอักษรและรูปต่างๆ

ซอร์สโค้ด (Source code) ของโปรแกรมกราฟิกเอดิเตอร์

โปรแกรมกราฟิกเอดิเตอร์ ประกอบด้วยไฟล์ 3 ไฟล์ คือ

1. var.h ซึ่งเป็น include file ใช้สำหรับเก็บสตริงเลเบล (string label) ของเมนูใช้งานต่างๆ และกำหนดโครงสร้างของบัฟเฟอร์ (buffer) ที่ใช้เก็บตำแหน่งและแอตทริบิวต์ (attributes) เพื่อใช้ในการรีดรอว์ (redraw)

2. gredit.c เป็น main program ที่ประกอบด้วยฟังก์ชันการทำงานที่ใช้ในการทำงานต่างๆ

3. thai.afb เป็นการสร้างฟอนต์ภาษาไทยตามรูปแบบที่กำหนดของ Adobe ASCII font bitmap

4. thai24.fb เป็นฟอนต์ภาษาไทยในลักษณะของ bitmap ที่เกิดจากการคอนเวอร์ทไฟล์ thai.afb ด้วยคำสั่ง convertfont



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
static char *menu_label[] = {
    "File",
    "Text",
    "Font",
    "Color",
    "Line",
    "Shapes",
    "Clear",
};
```

```
static char *file_label[] = {
    "Load",
    "Save",
    "Dir",
    "Quit"
};
```

```
static char *font_label[] = {
    "8x13",
    "8x13bold",
    "9x15",
    "thai24"
};
```

```
static char *color_label[] = {
    "Red",
    "Green",
    "Blue",
    "Yellow",
    "Black"
};
```

```
static char *line_label[] = {
    "Style",
    "Width"
};
```

```
static char *shapes_label[] = {
    "Point",
    "Line",
    "Circle",
    "N-tangle",
    "Rectangle"
};
```

```
static char *style_label[] = {
    "Solid",
    "Dash1",
    "Dash2"
};
```

```
static char *width_label[] = {
    " 1  ",
    " 2  ",
    " 3  "
};
```

```
struct pnt_buf {
    int x,y;
    unsigned long color;
};
```

```
struct line_buf {
    int x1,y1,x2,y2;
    unsigned long color;
    int style,l_width;
};
```

```
struct rec_buf {
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่อนุญาตให้นำไปใช้ในการเรียนการสอนโดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัย
หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูงและต้องอภัยถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int x,y;
int width,height;
unsigned long color;
int style,l_width;
} ;

struct txt_buf {
int x,y;
char str[200];
unsigned long color;
char *fnt;
} ;

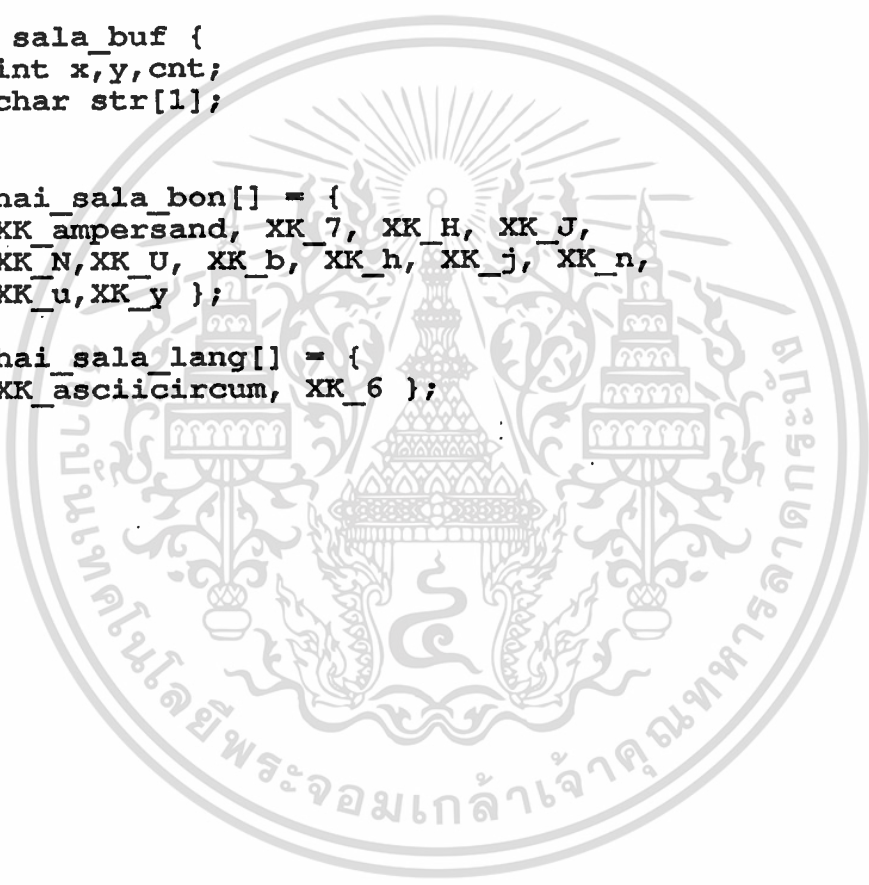
struct ntan_buf {
int x[50],y[50];
unsigned long color;
int style,l_width;
int num;
} ;

struct sala_buf {
int x,y,cnt;
char str[1];
} ;

int thai_sala_bon[] = {
XK_ampersand, XK_7, XK_H, XK_J,
XK_N,XK_U, XK_b, XK_h, XK_j, XK_n,
XK_u,XK_y };

int thai_sala_lang[] = {
XK_asciicircum, XK_6 };

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

027846



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <X11/Xlib.h>
#include <X11/Xutil.h>
#include <X11/Xos.h>
#include <X11/Xatom.h>
#include <X11/keysym.h>
#include <X11/cursorfont.h>
```

```
#include "X11/bitmaps/icon"
#include "/usr/staff/pirat/cross"
#include "/usr/staff/pirat/var.h"
```

```
#include <stdio.h>
#include <signal.h>
```

```
#define STYLE 1
#define WIDTH 0
#define MAX_CHOICE 7
#define MAX_COLORS 5
#define NONE 0
#define NOTDEFINED 0
#define BLACK 1 /* this is not a pixel value; this is a flag */
#define WHITE 0 /* this is not a pixel value; this is a flag */
#define HORIZON 1
#define VERTICAL 0
```

```
#define MAX_STR_LEN 256
#define MAX_BUF_LEN 10
```

```
Window focus_window;
Window inverted_pane = NONE;
Window inverted_fi = NONE;
Window inverted_txt = NONE;
Window inverted_fnt = NONE;
Window inverted_clr = NONE;
Window inverted_lin = NONE;
Window inverted_shp = NONE;
Window inverted_sty = NONE;
Window inverted_wth = NONE;
```

```
Display *display;
int screen;
GC gc, rgc, active_gc;
```

```
Visual *visual;
Colormap cmap;
Cursor xterm, cross;
```

```
Window mainwin, menuwin;
Window panes[MAX_CHOICE];
Window sub_fi, fi_panes[4];
Window sub_fnt, fnt_panes[4];
Window sub_clr, clr_panes[5];
Window sub_lin, lin_panes[2];
Window sub_shp, shp_panes[5];
Window sub_sty, sty_panes[3];
Window sub_wth, wth_panes[3];
```

```
KeySym keysym, oldchr, newchr;
Font font;
XEvent event;
XFontStruct *font_info;
XCharStruct overall;
XSizeHints hints;
XColor exact_def;
XSetWindowAttributes setwinattr;
XGCValues values;
```

ขอแจ้งการจ้างงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ทั้งนี้ ห้ามนำไปดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

XPoint    point1,point2;

int       menu_width, menu_height;
int       char_count;
int       colors[MAX_COLORS];
unsigned long foreground_pixel,background_pixel,border_pixel;
unsigned long valuemask;
char      *font_name = "8x13bold";
char      *string1 ;
char      *win_name = "Graphic Editor V 1.0";
char      *icon_name = "Graphic Editor";
char      *save_name = "Graphic:Store";
unsigned long common_color;
int       common_style,common_width;
char      *common_font;
int       shape_mode = 1;
int       dex, fno = 0;
int       how,oh;
int       chk_flg = 0;
int       sl_flg = 0;
struct pnt_buf pnt_buffer[200];
struct line_buf line_buffer[200];
struct rec_buf circle_buffer[200],rec_buffer[200];
struct txt_buf txt_buffer[200];
struct nta_buf nta_buffer[200];
struct sala_buf sala_buffer[200];
unsigned int line_count, cir_count, rec_count, pnt_count,txt_count;
unsigned int nta_count, sala_count,numcnt;
int cap_style = CapRound;
int join_style = JoinRound;
int x_str, y_str, org_x, org_y, count;

```

```

main(argc,argv)
int argc;
char **argv;
{
    Window    confine_to;
    Window    assoc_win = NOTDEFINED;
    Window    root, child;
    Pixmap    icon_pixmap, csource, cmask;
    XComposeStatus *compose;
    XWMHints  wmhints;
    XColor    cfore,cback;
    int       main_width, main_height;
    int       pane_height, pane_width;
    int       x , y , border_width,i,j ;
    int       winindex, width, height;
    int       direction, ascent, descent;
    int       cursor_shape;
    int       bufsize = MAX_BUF_LEN;
    int       length,len;
    int       txt,txt;
    int       pos_x, pos_y,w,h;
    int       wl,hl;
    int       prev_x, prev_y, fst_x, fst_y;
    int       root_x, root_y;
    char      string[MAX_STR_LEN];
    char      buffer[MAX_BUF_LEN];
    unsigned int button;
    unsigned int width_return, height_return;
    unsigned int keys_buttons;
    unsigned int c, flag = 0;

```

```

if (( display = XOpenDisplay(NULL)) == NULL)
{

```

```

(void) fprintf( stderr, "mainwin: cannot connect to \
                X server %s\n", XDisplayName(NULL));
    exit(-1);
}

/* SET COLOR,USED DEFAULT VALUES */
visual = DefaultVisual(display,screen);
cmap = DefaultColormap(display,screen);

for(i=0; i < MAX_COLORS; i++) {
    if (!XParseColor (display,cmap,color_label[i],&exact_def))
        {
            (void) fprintf(stderr,"color name %s not in database",
                color_label[i]);
            exit(-1);
        }
    if (!XAllocColor (display,cmap,&exact_def))
        {
            (void) fprintf(stderr,"all colorcells allocated and read/write\n");
            exit(-1);
        }
    colors[i] = exact_def.pixel;
}

/* SET PIXEL VALUES */
border_pixel = colors[0];
background_pixel = 0; /*background_pixel; */
foreground_pixel = colors[4];

screen = DefaultScreen(display);
common_color = BlackPixel(display, screen);
common_style = LineSolid;
common_width = 1;

/* ACCESS FONT */
font_info = XLoadQueryFont (display,font_name);
if (font_info == NULL)
    {
        (void) fprintf( stderr, "mainmenu: cannot open font %s\n", font_name);
        exit(-1);
    }

main_width = 800;  main_height = 600;

/* OPEN MAIN WINDOW */
mainwin = XCreateSimpleWindow(display,RootWindow(display,screen),
    x = 5, y = 7, main_width, main_height,
    border_width = 5,
    border_pixel,background_pixel);

hints.x = x; hints.y = y;
hints.height = main_height;
hints.width = main_width;
hints.min_height = 250;
hints.min_width = 350;
hints.max_height = 800;
hints.max_width = 1000;
hints.flags = PPosition | PSize | PMinSize | PMaxSize;

wmhints.input = True;
wmhints.flags = InputHint;
XSetWMHints(display,mainwin,&wmhints);
XSetStandardProperties(display,mainwin,win_name,icon_name,None,
    argv,argc,&hints);

string1 = menu_label[6];
char_count = strlen(string1) ; /* get length of the longest string */

```

เอกสารนี้เป็นเอกสารของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ควรกรณิใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือแจกจ่ายแก่บุคคลอื่นโดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัย

```

/* DETERMINED THE EXTEND OF EACH MENU PANE BASED ON THE FONT SIZE */
XTextExtents(font_info, string1, char_count, &direction, &ascent,
             &descent, &overall);
menu_height = overall.ascent + overall.descent + 15;
menu_width = main_width;
pane_width = 800/MAX_CHOICE-8;
pane_height = menu_height-12;

/* CREATE OPAQUE WINDOW */
menuwin = XCreateSimpleWindow(display,mainwin,0,0,menu_width,menu_height,
                             border_width = 0,border_pixel,
                             background_pixel);

/* CREATE THE CHOICE WINDOWS FOR THE TEXT */
for (winindex = 0; winindex < MAX_CHOICE; winindex++)
{
    panes[winindex] = XCreateSimpleWindow(display,menuwin,
        800/MAX_CHOICE*winindex+2, 2, 800/MAX_CHOICE-8,
        pane_height, border_width = 3 ,
        border_pixel,
        background_pixel);
    XSelectInput (display, panes[winindex], ButtonPressMask |
        ButtonReleaseMask | ExposureMask);
}

XSelectInput (display, mainwin, ExposureMask | ButtonPressMask |
    ButtonReleaseMask | PointerMotionHintMask |
    KeyPressMask | EnterWindowMask | LeaveWindowMask |
    SubstructureNotifyMask | ButtonMotionMask

create_gc();
XSetFont (display,gc,font_info->fid);
XSetFont (display,rgc,font_info->fid);

font = XLoadFont (display,"8x13");
common_font = "8x13";
XSetFont (display,active_gc,font);
txt_buffer[txt_count].fnt = font_label[0];

create_win();

/* THESE DON'T APPEAR UNTIL PARENT (MAINWIN) IS MAPPED */
XMapSubwindows (display,menuwin);

cross = XCreateFontCursor (display, XC_cross);
XDefineCursor (display,mainwin,cross);
xterm = XCreateFontCursor (display, XC_xterm);

focus_window = mainwin;

/* MAP THE MENU WINDOW (AND ITS SUBWINDOWS) TO THE SCREEN */
XMapWindow (display, mainwin);
XMapWindow (display,menuwin);
buffer[1] = 0;
/* LOOP GETTING EVENTS ON THE MENU WINDOW */
while (1)
{
    /* WAIT FOR AN EVENT */
    XNextEvent (display,&event);
    switch (event.type) {
        case Expose:
            if (event.xexpose.window == mainwin) {
                while (XCheckTypedWindowEvent (display,mainwin, Expose, &event));
                XClearWindow (display, mainwin);
            }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ หากมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (line_count != 0)
    for (c=0; c < line_count+1; c++) {
        XSetForeground(display, active_gc, line_buffer[c].color);
        XSetLineAttributes(display, active_gc, line_buffer[c].l_width,
            line_buffer[c].style, cap_style, join_style);
        XDrawLine(display, mainwin, active_gc,
            line_buffer[c].x1,
            line_buffer[c].y1, line_buffer[c].x2,
            line_buffer[c].y2);
    }
if (pnt_count != 0)
    for (c=0; c < pnt_count+1; c++) {
        XSetForeground(display, active_gc, pnt_buffer[c].color);
        XDrawPoint(display, mainwin, active_gc, pnt_buffer[c].x,
            pnt_buffer[c].y);
    }
if (cir_count != 0)
    for (c=0; c < cir_count+1; c++) {
        XSetForeground(display, active_gc, circle_buffer[c].color);
        XSetLineAttributes(display, active_gc, circle_buffer[c].l_width,
            circle_buffer[c].style, cap_style, join_style);
        XDrawArc(display, mainwin, active_gc, circle_buffer[c].x,
            circle_buffer[c].y, circle_buffer[c].width,
            circle_buffer[c].height, 0, 360*64);
    }
if (rec_count != 0)
    for (c=0; c < rec_count+1; c++) {
        XSetForeground(display, active_gc, rec_buffer[c].color);
        XSetLineAttributes(display, active_gc, rec_buffer[c].l_width,
            rec_buffer[c].style, cap_style, join_style);
        XDrawRectangle(display, mainwin, active_gc, rec_buffer[c].x,
            rec_buffer[c].y, rec_buffer[c].width,
            rec_buffer[c].height);
    }
if (txt_count != 0)
    for (c=0; c < txt_count+1; c++) {
        XSetForeground(display, active_gc, txt_buffer[c].color);
        font = XLoadFont(display, txt_buffer[c].fnt);
        XSetFont(display, active_gc, font);
        XDrawString(display, mainwin, active_gc, txt_buffer[c].x,
            txt_buffer[c].y, txt_buffer[c].str,
            strlen(txt_buffer[c].str));
        for (i=0; i < sala_count+1; i++)
            if (sala_buffer[i].cnt == c)
                XDrawString(display, mainwin, active_gc, sala_buffer[i].x,
                    sala_buffer[i].y, sala_buffer[i].str, 1);
    }
if (ntan_count != 0)
    for (c=0; c < ntan_count+1; c++) {
        XSetForeground(display, active_gc, ntan_buffer[c].color);
        XSetLineAttributes(display, active_gc, ntan_buffer[c].l_width,
            ntan_buffer[c].style, cap_style, join_style);
        for (i=0; i < ntan_buffer[c].num; i++)
            XDrawLine(display, mainwin, active_gc, ntan_buffer[c].x[i],
                ntan_buffer[c].y[i], ntan_buffer[c].x[i+1],
                ntan_buffer[c].y[i+1]);

        XDrawLine(display, mainwin, active_gc,
            ntan_buffer[c].x[i],
            ntan_buffer[c].y[i],
            ntan_buffer[c].x[0], ntan_buffer[c].y[0]);
        XSetForeground(display, active_gc, common_color);
        XSetLineAttributes(display, active_gc, common_width,
            common_style, cap_style, join_style);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีเมล: office@scs.ac.th หรือ โทร: 0-2562-03100-101011
 ที่มีการนำไปใช้

```

else {
if (assoc_win == event.xexpose.window) /* if icon */
    XDrawString(display,assoc_win,gc,2,ascent+2,icon_name,
                strlen(icon_name));
else /* it is a pane, might be inverted */
{
    if (inverted_pane == event.xexpose.window)
        paint_pane(event.xexpose.window, panes,menu_label, gc,rgc,
                    font_info,BLACK,HORIZON);
    else
        paint_pane(event.xexpose.window,panes,menu_label, gc, rgc,
                    font_info,WHITE,HORIZON);
}
}
break;

case ButtonRelease:
    break;

case ButtonPress:
    if (event.xbutton.window == mainwin) {
        if (event.xbutton.button == Button1) {
            switch (shape_mode) {
                case 0:
                    point1.x = event.xbutton.x;
                    point1.y = event.xbutton.y;

                    pnt_buffer[pnt_count].x = point1.x;
                    pnt_buffer[pnt_count].y = point1.y;
                    pnt_count++;

                    XDrawPoint(display,mainwin,active_gc,point1.x,
                                point1.y);
                    break;
                case 1:
                    dex++;
                    if (dex == 1) {
                        point1.x = event.xbutton.x;
                        point1.y = event.xbutton.y;

                        line_buffer[line_count].x1 = point1.x;
                        line_buffer[line_count].y1 = point1.y;

                        prev_x = point1.x;
                        prev_y = point1.y;
                    }
                    else
                    if (dex == 2) {
                        point2.x = event.xbutton.x;
                        point2.y = event.xbutton.y;
                        /* save point in buffer */
                        line_buffer[line_count].x2 = point2.x;
                        line_buffer[line_count].y2 = point2.y;
                        line_buffer[line_count].color = common_color;
                        line_buffer[line_count].style = common_style;
                        line_buffer[line_count].l_width = common_width;
                        line_count++;

                        XDrawLine(display,mainwin,active_gc,point1.x,
                                    point1.y,point2.x,point2.y);
                        XDrawLine(display, mainwin, active_gc, point1.x,
                                    point1.y, prev_x, prev_y);
                        dex = 0;
                    }
                    break;
                case 2:
                    dex++;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาของนักศึกษาชั้นปริญญาตรี
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแต่งเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (dex == 1) {
    point1.x = event.xbutton.x;
    point1.y = event.xbutton.y;

    circle_buffer[cir_count].x = point1.x;
    circle_buffer[cir_count].y = point1.y;

    prev_x = point1.x;
    prev_y = point1.y;
}
else
if (dex == 2) {
    point2.x = event.xbutton.x;
    point2.y = event.xbutton.y;
    w = point2.x - point1.x;
    h = point2.y - point1.y;

    circle_buffer[cir_count].width = w;
    circle_buffer[cir_count].height = h;
    circle_buffer[cir_count].color = common_color;
    circle_buffer[cir_count].style = common_style;
    circle_buffer[cir_count].l_width = common_width;
    cir_count++;

    XDrawArc(display,mainwin,active_gc,point1.x,
              point1.y,w,h,0,360*64);
    XDrawArc(display,mainwin,active_gc,point1.x,
              point1.y,prev_x-point1.x,
              prev_y-point1.y,0,360*64);
    dex = 0;
}
break;
case 3:
    dex++;
    how = 0;
    if (dex == 1) {
        point1.x = event.xbutton.x;
        point1.y = event.xbutton.y;
        prev_x = point1.x;
        prev_y = point1.y;
        fst_x = point1.x;
        fst_y = point1.y;
        ntan_buffer[ntan_count].x[numcnt] = point1.x;
        ntan_buffer[ntan_count].y[numcnt] = point1.y;
        numcnt++;
    }
    else
    if (dex > 1) {
        point2.x = event.xbutton.x;
        point2.y = event.xbutton.y;

        ntan_buffer[ntan_count].x[numcnt] = point2.x;
        ntan_buffer[ntan_count].y[numcnt] = point2.y;
        numcnt++;

        XDrawLine(display,mainwin,active_gc,point1.x,
                  point1.y,point2.x,point2.y);
        XDrawLine(display,mainwin,active_gc,point1.x,
                  point1.y,prev_x,prev_y);
        point1.x = point2.x;
        point1.y = point2.y;
        prev_x = point1.x;
        prev_y = point1.y;
    }
break;
case 4:
    dex++;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

/*WAIT FOR RELEASE ; IF ON CORRECT BUTTON EXIT */
XMaskEvent(display, ButtonReleaseMask, &event);
if (event.xbutton.button == button)
    break;
}

/* IF PRESS AND RELEASE OCCURED IN SAME WINDOW, DO COMMAND
* IF NOT , DO NOTHING */
if (inverted_pane == event.xbutton.window)
{
    /* CONVERT WINDOW ID TO WINDOW ARRAY INDEX */
    for (winindex = 0; inverted_pane != panes[winindex]; winindex++);
    switch (winindex)
    {
        case 0:
            file_menu(sub_fi, fi_panes, file_label);
            break;
        case 1:
            if (oh == 0) {
                oh = 1;
                XDefineCursor(display, mainwin, xterm);
            }
            else {
                oh = 0;
                XDefineCursor(display, mainwin, cross);
            }
            break;
        case 2:
            font_menu(sub_fnt, fnt_panes, font_label);
            break;
        case 3:
            color_menu(sub_clr, clr_panes, color_label);
            break;
        case 4:
            line_menu(sub_lin, lin_panes, line_label);
            break;
        case 5:
            shapes_menu(sub_shp, shp_panes, shapes_label);
            break;
        case 6:
            XClearWindow(display, mainwin);
            for (c=0; c < 200 ; c++) {
                line_buffer[c].x1 = 0;
                line_buffer[c].y1 = 0;
                line_buffer[c].x2 = 0;
                line_buffer[c].y2 = 0;
                circle_buffer[c].x = 0;
                circle_buffer[c].y = 0;
                circle_buffer[c].width = 0;
                circle_buffer[c].height = 0;
                rec_buffer[c].x = 0;
                rec_buffer[c].y = 0;
                rec_buffer[c].width = 0;
                rec_buffer[c].height = 0;
                pnt_buffer[c].x = 0;
                pnt_buffer[c].y = 0;
                for (i=0; i < ntan_buffer[c].num; i++) {
                    ntan_buffer[c].x[i] = 0;
                    ntan_buffer[c].y[i] = 0;
                }
            }
            ntan_buffer[c].num = 0;
            txt_buffer[c].x = 0;
            txt_buffer[c].y = 0;
            txt_buffer[c].str[0] = 0;
            for (i=0; i < sala_count+1; i++) {
                sala_buffer[i].x = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิ... ไปถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        sala_buffer[i].y = 0;
        sala_buffer[i].cnt = 0;
        sala_buffer[i].str[0] = 0;
    }
}

line_count = 0;
rec_count = 0;
cir_count = 0;
pnt_count = 0;
txt_count = 0;
sala_count = 0;
string[0] = 0;
break;

default:
    (void) fprintf(stderr, "Something went wrong\n");
    break;
} /* END SWITCH */
} /* END IF */
/* INVERT BACK HERE (LOGICAL FUNCTION IS COPY) */
paint_pane(inverted_pane, panes, menu_label, gc,
            rgc, font_info, WHITE, HORIZON);
inverted_pane = NONE;
XFlush(display);
}
break;

case KeyPress:
    if (oh == 1) {
        XSetInputFocus(display, mainwin, RevertToParent, CurrentTime);
        if (event.xkey.window == mainwin) {
            count = XLookupString(&event, buffer, bufsize,
                                   &keysym, &compose);
            txt_buffer[txt_count].x = org_x+18;
            txt_buffer[txt_count].y = org_y;
            newchr = keysym;
            if (((oldchr == XK_b) || (oldchr == XK_6)) && ((newchr == XK_b) ||
                (newchr == XK_6) || (newchr == XK_7) || (newchr == XK_n) ||
                (newchr == XK_y) || (newchr == XK_ampersand) || (newchr == XK_u) ||
                (newchr == XK_asciicircum)))
                XBell(display, 100);
            else
                if (((oldchr == XK_7) || (oldchr == XK_n) || (oldchr == XK_y)) &&
                    ((newchr == XK_N) || (newchr == XK_F) || (newchr == XK_w) ||
                    (newchr == XK_g) || (newchr == XK_c) || (newchr == XK_period) ||
                    (newchr == XK_6) || (newchr == XK_7) || (newchr == XK_n) ||
                    (newchr == XK_y) || (newchr == XK_ampersand) || (newchr == XK_u) ||
                    (newchr == XK_asciicircum) || (newchr == XK_b)))
                        XBell(display, 100);
                else
                    if (((oldchr == XK_j) || (oldchr == XK_h) || (oldchr == XK_U) ||
                        (oldchr == XK_J) || (oldchr == XK_H)) && ((newchr == XK_j) ||
                        (newchr == XK_h) || (newchr == XK_U) || (newchr == XK_J) ||
                        (newchr == XK_ampersand) || (newchr == XK_u) ||
                        (newchr == XK_asciicircum) || (newchr == XK_b) || (newchr == XK_6)
                        || (newchr == XK_H) || (newchr == XK_N)))
                            XBell(display, 100);
                    else
                        if (((oldchr == XK_u) || (oldchr == XK_asciicircum)) &&
                            ((newchr == XK_N) || (newchr == XK_asciicircum) ||
                            (newchr == XK_6) || (newchr == XK_7) || (newchr == XK_n) ||
                            (newchr == XK_y) || (newchr == XK_ampersand) || (newchr == XK_u) ||
                            (newchr == XK_b)))
                                XBell(display, 100);
                            else
                                if (((oldchr == XK_N) || (oldchr == XK_F) || (oldchr == XK_w) ||
                                    (oldchr == XK_period) || (oldchr == XK_g) || (oldchr == XK_c) ||

```

เอกสารนี้เป็นเอกสารของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามเผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ว่ากรณีใดๆ ทั้งสิ้น ขอ Bell(display, 100); และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(oldchr == XK_ampersand) && ((newchr == XK_H) || (newchr == XK_j)
|| (newchr == XK_h) || (newchr == XK_U) || (newchr == XK_J) ||
(newchr == XK_N) || (newchr == XK_F) || (newchr == XK_w) ||
(newchr == XK_g) || (newchr == XK_c) || (newchr == XK_period) ||
(newchr == XK_6) || (newchr == XK_7) || (newchr == XK_n) ||
(newchr == XK_y) || (newchr == XK_ampersand) || (newchr == XK_u) ||
(newchr == XK_asciicircum) || (newchr == XK_b)))
XBell(display,100);

```

```

else {

```

```

    if ((keySYM == XK_Return) || (keySYM == XK_KP_Enter)
        || (keySYM == XK_Linefeed)) {
        for (i = 0; i < 200 ; i++)
            txt_buffer[txt_count].str[i] = string[i];
            txt_buffer[txt_count].color = common_color;
            txt_count++;
            txt_buffer[txt_count].fnt = common_font;
            string[0] = 0;
            x_str = pos_x;
            if (fno == 3) {
                y_str = y_str + 50;
                org_y = y_str;
            }
            else {
                y_str = y_str + font_info->max_bounds.ascent +
                    font_info->max_bounds.descent + 4;
                org_y = y_str;
            }
        }
        else
            if (((keySYM >= XK_KP_Space) && (keySYM <= XK_KP_9)) ||
                ((keySYM >= XK_space) && (keySYM <= XK_asciitilde))) {
                if ((strlen(string) + strlen(buffer)) >= MAX_STR_LEN)
                    XBell(display,100);
                else {
                    chk_loc();
                    if (sl_flg == 1)
                        sl_flg = 0;
                    else
                        strcat(string,buffer);
                }
            }
        else if ((keySYM >= XK_Shift_L) &&
                (keySYM <= XK_Hyper_R))
            break ;
        else if ((keySYM >= XK_F1) && (keySYM <= XK_F12))
            if (buffer == NULL)
                printf("Unmapped function key\n");
            else if ((strlen(string) + strlen(buffer))
                    >= MAX_STR_LEN)
                XBell(display,100);
            else {
                chk_loc();
                if (sl_flg == 1)
                    sl_flg = 0;
                else
                    strcat(string,buffer);
            }
        else if ((keySYM == XK_BackSpace) || (XK_Delete)) {
            if ((length = strlen(string)) > 0) {
                string[length-1] = NULL;
                x_str = x_str - font_info->max_bounds.width;
                XDrawString(display, mainwin, active_gc,
                    x_str,y_str,"",1);
            }
            else
                XBell(display,100);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาและการวิจัยเท่านั้น การนำเอกสารนี้ไปใช้ในการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงที่มาทุกครั้ง

```

}

```

```

else

```

```

    XBell(display,100);

```

```

    }
    else {
        printf("keysym %s is not handle\n",
            XKeysymToString(keysym));
        XBell(display,100);
    }
    if ( fno == 3) {
        for (i=0; i<12; i++) {
            if (keysym == thai_sala_bon[i])
                flag = 1;
        }
        for (i=0; i<2; i++) {
            if (keysym == thai_sala_lang[i])
                flag = 2;
        }
        if (flag == 1) {
            sala_buffer[sala_count].x = x_str;

            if ((oldchr == XK_b) || (oldchr == XK_n) ||
                (oldchr == XK_u) || (oldchr == XK_y) ||
                (oldchr == XK_7)) {
                sala_buffer[sala_count].y = y_str-30;
                XDrawImageString(display, mainwin, active_gc,
                    x_str,y_str-30,buffer, strlen(buffer));
            }
            else {
                sala_buffer[sala_count].y = y_str-20;
                XDrawImageString(display, mainwin, active_gc,
                    x_str,y_str-20,buffer, strlen(buffer));
            }
            oldchr = keysym;
            sala_count++;
            flag = 0;
        }
        else
            if (flag == 2) {
                sala_buffer[sala_count].x = x_str+2;
                sala_buffer[sala_count].y = y_str+23;
                oldchr = keysym;
                sala_count++;
                XDrawString(display, mainwin, active_gc,
                    x_str+2,y_str+23, buffer, strlen(buffer));
                flag = 0;
            }
            else {
                oldchr = keysym;
                XDrawImageString(display,mainwin,active_gc,x_str,
                    y_str,buffer, strlen(buffer));
            }
    }
    else
        XDrawImageString(display,mainwin,active_gc,
            x_str, y_str, buffer, strlen(buffer));
}
}
}

break;
case MotionNotify:
    if (event.xmotion.window == mainwin) {
        while (XCheckMaskEvent(display, ButtonMotionMask, &event));
        if (!XQueryPointer(display, event.xmotion.window, &root, &child,
            &root_x, &root_y, &pos_x, &pos_y, &keys_buttons))
            break;
        x_str = org_x = pos_x;
        y_str = org_y = pos_y;
        string[0] = 0;
        switch(shape_mode) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น การนำออกใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

ไม่ว่ากรณีใดๆ ทั้งสิ้น ขอสงวนสิทธิ์ในสิ่งที่ปรากฏ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 0:
    break;
case 1:
    if (dex == 1) {
        XDrawLine(display,mainwin,active_gc,point1.x,point1.y,
            prev_x,prev_y);
        XDrawLine(display,mainwin,active_gc,point1.x,point1.y,
            pos_x, pos_y);
        prev_x = pos_x;
        prev_y = pos_y;
    }
    break;
case 2:
    w1 = pos_x - point1.x;
    h1 = pos_y - point1.y;
    if (dex == 1) {
        XDrawArc(display, mainwin, active_gc, point1.x, point1.y,
prev_x-point1.x, prev_y-point
        XDrawArc(display, mainwin, active_gc, point1.x, point1.y,
w1, h1, 0, 360 * 64);
        prev_x = pos_x;
        prev_y = pos_y;
    }
    break;
case 3:
    if (dex >= 1) {
        XDrawLine(display,mainwin,active_gc,point1.x,point1.y,
            prev_x, prev_y);
        XDrawLine(display, mainwin, active_gc, point1.x, point1.y,
            pos_x, pos_y);
        prev_x = pos_x;
        prev_y = pos_y;
    }
    break;
case 4:
    w1 = pos_x - point1.x;
    h1 = pos_y - point1.y;
    if (dex == 1) {
        XDrawRectangle(display,mainwin,active_gc,point1.x, point1.y,
            prev_x-point1.x, prev_y-point1.y);
        XDrawRectangle(display, mainwin, active_gc,
            point1.x, point1.y, w1, h1);
        prev_x = pos_x;
        prev_y = pos_y;
    }
    break;
}
}
break;
case DestroyNotify:
    XCloseDisplay(display);
    break;
case EnterNotify:
    break;
case LeaveNotify:
    break;
case CirculateNotify:
case ConfigureNotify:
    /* if (event.xconfigure.window == mainwin)
    {
        width = event.xconfigure.width;
        height = event.xconfigure.height;
        menu_width = width;
        XResizeWindow(display, menuwin, menu_width, menu_height);
        XRaiseWindow(display, menuwin);
    } */

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีก และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    case UnmapNotify:
        /* ALL THESE UNCOVER AREAS OF SCREEN */
        break;
    case CreateNotify:
    case GravityNotify:
    case MapNotify:
    case ReparentNotify:
        /* DON'T NEED THESE BUT GET THEM AUTOMATICALLY
         * SINCE WE NEED DESTROYNOTIFY AND UNMAPNOTIFY */
        break;
    default:
        fprintf(stderr, "winman: got unexpected event %s\n", event.type);
    } /* END SWITCH */
} /* END MAIN LOOP */
} /* END MAIN */

```

```

chk_loc()
{
    int i;
    int chk_flg=0;

    if (fno != 3)
        x_str = x_str + font_info->max_bounds.width ;
    else {
        for(i=0; i<12; i++) {
            if (keysym == thai_sala_bon[i]) {
                sala_buffer[sala_count].str[0] = thai_sala_bon[i];
                sala_buffer[sala_count].cnt = txt_count;
                chk_flg = 1;
                sl_flg = 1;
            }
        }
        for (i=0; i<2; i++) {
            if (keysym == thai_sala_lang[i]) {
                sala_buffer[sala_count].str[0] = thai_sala_lang[i];
                sala_buffer[sala_count].cnt = txt_count;
                chk_flg = 2;
                sl_flg = 1;
            }
        }
        if ((chk_flg == 1) || (chk_flg == 2))
            chk_flg = 0;
        else
            x_str = x_str + font_info->max_bounds.width + 9 ;
    }
}

```

```

paint_pane(window, panes, menulabel, ngc, rgc, font_info, mode, hv_mode)
Window window;
Window panes[];
char *menulabel[];
GC ngc,rgc;
XFontStruct *font_info;
int mode;
int hv_mode;
{
    int win;
    int y, x;
    GC p_gc;
    if (mode == BLACK) {
        XSetWindowBackground(display, window, BlackPixel(display, screen));
        p_gc = rgc;
    }
    else {

```

เอกสารนี้เป็นทรัพย์สินของสำนักงานคณะกรรมการการศึกษาระดับมัธยมศึกษาตอนต้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ XSetWindowBackground(display, window, BlackPixel(display, screen));

```

        XSetWindowBackground(display, window, WhitePixel(display, screen));
        p_gc = ngc;
    }
    /* CLEARING REPAINTS THE BACKGROUND */
    XClearWindow(display, window);

    /* FIND OUT INDEX OF WINDOW FOR LABEL TEXT */
    for (win = 0; window != panes[win]; win++)
        ;
    if (hv_mode == HORIZON) {
        x = font_info->min_bounds.width;
        y = 10;
    }
    else
    {
        x = 0;
        y = font_info->max_bounds.ascent;
    }

    /* THE STRING LENGHT IS NECCESSARY BECAUSE STRINGS FOR XDRAWSTRING
    * MAY NOT BE NULL TERMINATED */
    XDrawString(display, window, p_gc, x, y, menulabel[win],
                strlen( menulabel[win]));
}

file menu(sub_fi, file_panes, file_label)
Window sub_fi;
Window file_panes[];
char *file_label[];
{
    int winindex;
    int button;
    int i;
    XMapSubwindows(display, sub_fi);
    XMapWindow(display, sub_fi);

    while (1)
    {
        XNextEvent(display, &event);
        switch(event.type) {
            case Expose:
                if (inverted_fi == event.xexpose.window)
                    paint_pane(event.xexpose.window, file_panes, file_label, gc,
                                rgc, font_info, BLACK, VERTICAL);
                else
                    paint_pane(event.xexpose.window, file_panes, file_label, gc,
                                rgc, font_info, WHITE, VERTICAL);
                break;
            case ButtonPress:
                if (event.xbutton.window == mainwin) {
                    XUnmapSubwindows(display, sub_fi);
                    XUnmapWindow(display, sub_fi);
                    return;
                }
                else
                    if ((event.xbutton.window == panes[0]) ||
                        (event.xbutton.window == panes[1]) ||
                        (event.xbutton.window == panes[2]) ||
                        (event.xbutton.window == panes[3]) ||
                        (event.xbutton.window == panes[4]) ||
                        (event.xbutton.window == panes[5]) ||
                        (event.xbutton.window == panes[6])) {
                            XUnmapSubwindows(display, sub_fi);
                            XUnmapWindow(display, sub_fi);
                            return;
                        }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามทำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีเมล: office@kmutt.ac.th; จำของเอกสารทุกครั้งที่มีการนำไปใช้

```

else {
    paint_pane(event.xbutton.window, file_panes, file_label, gc,
               rgc, font_info, BLACK, VERTICAL);
    button = event.xbutton.button;
    inverted_fi = event.xbutton.window;

while (1)
    {
        while (XCheckTypedEvent(display, ButtonPress, &event));
        XMaskEvent(display, ButtonReleaseMask, &event);
        if (event.xbutton.button == button)
            break;
    }
    if (inverted_fi == event.xbutton.window)
        {
            for (winindex = 0; inverted_fi != file_panes[winindex]; winindex++)
                ;
            switch(winindex) {
                case 0:
                    break;
                case 1:
                    break;
                case 2:
                    break;
                case 3:
                    XClearWindow(display, mainwin);
                    XFlush(display);
                    XCloseDisplay(display);
                    exit(1);
                    break;
            }
            paint_pane(inverted_fi, file_panes, file_label, gc, rgc,
                      font_info, WHITE, VERTICAL);
            inverted_fi = NONE;
            XFlush(display);
        }
        break;

    case DestroyNotify:
        break;
    case UnmapNotify:
        break;
    default:
        ;
    }
}
}

```

```

font_menu(sub_fnt, font_panes, font_label)
Window sub_fnt;
Window font_panes[];
char *font_label[];
{
    int winindex;
    int button;
    int i;
    XMapSubwindows(display, sub_fnt);
    XMapWindow(display, sub_fnt);

```

```

while (1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ switch(event.type) {
 case Expose:
 if (inverted_fnt == event.xexpose.window)
 paint_pane(event.xexpose.window, font_panes, font_label, gc,

```

        rgc, font_info, BLACK, VERTICAL);
    else
        paint_pane(event.xexpose.window, font_panes, font_label, gc,
                   rgc, font_info, WHITE, VERTICAL);
    break;
case ButtonPress:
    if (event.xbutton.window == mainwin) {
        XUnmapSubwindows(display, sub_fnt);
        XUnmapWindow(display, sub_fnt);
        return;
    }
    else
        if ((event.xbutton.window == panes[0]) ||
            (event.xbutton.window == panes[1]) ||
            (event.xbutton.window == panes[2]) ||
            (event.xbutton.window == panes[3]) ||
            (event.xbutton.window == panes[4]) ||
            (event.xbutton.window == panes[5]) ||
            (event.xbutton.window == panes[6])) {
            XUnmapSubwindows(display, sub_fnt);
            XUnmapWindow(display, sub_fnt);
            return;
        }
        else {
            paint_pane(event.xbutton.window, font_panes, font_label, gc,
                       rgc, font_info, BLACK, VERTICAL);
            button = event.xbutton.button;
            inverted_fnt = event.xbutton.window;

            while (1)
            {
                while (XCheckTypedEvent(display, ButtonPress, &event));
                XMaskEvent(display, ButtonReleaseMask, &event);
                if (event.xbutton.button == button)
                    break;
            }
            if (inverted_fnt == event.xbutton.window)
            {
                load_font(event.xbutton.window, font_panes, active_gc);
            }
            paint_pane(inverted_fnt, font_panes, font_label, gc,
                       rgc, font_info, WHITE, VERTICAL);
            inverted_fnt = NONE;
            XFlush(display);
        }
        break;

case DestroyNotify:
    break;
case UnmapNotify:
    break;
default:
    ;
}
}
}

```

```

color_menu(sub_clr, color_panes, color_label)
Window sub_clr;
Window color_panes[];
char *color_label[];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int winindex;
int button;
int i, win;
XMapSubwindows(display, sub_clr);
XMapWindow(display, sub_clr);

```

```

while (1)
{
    XNextEvent(display, &event);
    switch(event.type) {
    case Expose:
        if (inverted_clr == event.xexpose.window)
            paint_pane(event.xexpose.window, color_panes,color_label, gc,
                rgc, font_info, BLACK, VERTICAL);
        else
            paint_pane(event.xexpose.window, color_panes,color_label, gc,
                rgc, font_info, WHITE, VERTICAL);
        break;
    case ButtonPress:
        if (event.xbutton.window == mainwin) {
            XUnmapSubwindows(display,sub_clr);
            XUnmapWindow(display,sub_clr);
            return;
        }
        else
            if ((event.xbutton.window == panes[0]) ||
                (event.xbutton.window == panes[1]) ||
                (event.xbutton.window == panes[2]) ||
                (event.xbutton.window == panes[3]) ||
                (event.xbutton.window == panes[4]) ||
                (event.xbutton.window == panes[5]) ||
                (event.xbutton.window == panes[6])) {
                XUnmapSubwindows(display,sub_clr);
                XUnmapWindow(display,sub_clr);
                return;
            }
            else {
                paint_pane(event.xbutton.window, color_panes,color_label, gc,
                    rgc, font_info, BLACK, VERTICAL);
                button = event.xbutton.button;
                inverted_clr = event.xbutton.window;
                while (1)
                {
                    while (XCheckTypedEvent(display, ButtonPress, &event));
                    XMaskEvent(display, ButtonReleaseMask, &event);
                    if (event.xbutton.button == button)
                        break;
                }
                if (inverted_clr == event.xbutton.window)
                {
                    win = 0;
                    while (event.xbutton.window != clr_panes[win])
                        win ++;
                    common_color = colors[win];
                    XSetForeground(display, active_gc, common_color);
                }
                paint_pane(inverted_clr, color_panes,color_label, gc,
                    rgc, font_info, WHITE, VERTICAL);
                inverted_clr = NONE;
                XFlush(display);
            }
            break;

    case DestroyNotify:
        break;
    case UnmapNotify:
        break;
    default:
        ;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ว่ากรณีใดๆ ทั้งสิ้น ผู้ที่นำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตให้เข้าไปใช้ประโยชน์ด้านการค้า
 และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

line_menu(sub_lin, line_panes, line_label)
Window sub_lin;
Window line_panes[];
char *line_label[];
{
    int winindex;
    int button;
    int i;
    XMapSubwindows(display, sub_lin);
    XMapWindow(display, sub_lin);

    while (1)
    {
        XNextEvent(display, &event);
        switch(event.type) {
        case Expose:
            if (inverted_lin == event.xexpose.window)
                paint_pane(event.xexpose.window, line_panes, line_label, gc,
                            rgc, font_info, BLACK, VERTICAL);
            else
                paint_pane(event.xexpose.window, line_panes, line_label, gc,
                            rgc, font_info, WHITE, VERTICAL);
            break;
        case ButtonPress:
            if (event.xbutton.window == mainwin) {
                XUnmapSubwindows(display, sub_lin);
                XUnmapWindow(display, sub_lin);
                return;
            }
            else
                if ((event.xbutton.window == panes[0]) ||
                    (event.xbutton.window == panes[1]) ||
                    (event.xbutton.window == panes[2]) ||
                    (event.xbutton.window == panes[3]) ||
                    (event.xbutton.window == panes[4]) ||
                    (event.xbutton.window == panes[5]) ||
                    (event.xbutton.window == panes[6])) {
                    XUnmapSubwindows(display, sub_lin);
                    XUnmapWindow(display, sub_lin);
                    return;
                }
            else {
                paint_pane(event.xbutton.window, line_panes, line_label, gc,
                            rgc, font_info, BLACK, VERTICAL);
                button = event.xbutton.button;
                inverted_lin = event.xbutton.window;

                while (1)
                {
                    while (XCheckTypedEvent(display, ButtonPress, &event));
                    XMaskEvent(display, ButtonReleaseMask, &event);
                    if (event.xbutton.button == button)
                        break;
                }
            }
            if (inverted_lin == event.xbutton.window)
            {
                for (winindex = 0; inverted_lin != line_panes[winindex]; winindex++);
                switch (winindex)
                {
                    case 0:
                        i = style_menu(sub_sty, sty_panes, style_label);
                        if ((i == 1) || (i == 2)) {
                            inverted_lin = NONE;
                            XUnmapSubwindows(display, sub_lin);
                            XUnmapWindow(display, sub_lin);
                        }
                    }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ทำซ้ำ, แก้ไข, แจกจ่าย, หรือเผยแพร่เอกสารทุกครั้งที่มีการนำไปใช้

```

        return;
    }
    break ;
case 1:
    width_menu(sub_wth, wth_panes, width_label);
    inverted_lin = NONE;
    XUnmapSubwindows(display, sub_lin);
    XUnmapWindow(display, sub_lin);
    return;
    break;
}
}
paint_pane(inverted_lin, line_panes, line_label, gc,
           rgc, font_info, WHITE, VERTICAL);
inverted_lin = NONE;
XFlush(display);
}
break;

case DestroyNotify:
    break;
case UnmapNotify:
    break;
default:
    ;
}
}
}

```

```

shapes_menu(sub_shp, shapes_panes, shapes_label)
Window sub_shp;
Window shapes_panes[];
char *shapes_label[];
{
    int winindex;
    int button;
    int i;
    XMapSubwindows(display, sub_shp);
    XMapWindow(display, sub_shp);

    while (1)
    {
        XNextEvent(display, &event);
        switch(event.type) {
            case Expose:
                if (inverted_shp == event.xexpose.window)
                    paint_pane(event.xexpose.window, shapes_panes, shapes_label,
                               gc, rgc, font_info, BLACK, VERTICAL);
                else
                    paint_pane(event.xexpose.window, shapes_panes, shapes_label,
                               gc, rgc, font_info, WHITE, VERTICAL);
                break;
            case ButtonPress:
                if (event.xbutton.window == mainwin) {
                    XUnmapSubwindows(display, sub_shp);
                    XUnmapWindow(display, sub_shp);
                    return;
                }
                else
                    if ((event.xbutton.window == panes[0]) ||
                        (event.xbutton.window == panes[1]) ||
                        (event.xbutton.window == panes[2]) ||
                        (event.xbutton.window == panes[3]) ||
                        (event.xbutton.window == panes[4]) ||
                        (event.xbutton.window == panes[5]) ||
                        (event.xbutton.window == panes[6])) {
                            XUnmapSubwindows(display, sub_shp);
                    }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้ภายในเท่านั้น ห้ามเผยแพร่ไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งเอกสารทุกครั้งที่มีการนำไปใช้

```

        XUnmapWindow(display, sub_shp);
        return;
    }
    else {
        paint_pane(event.xbutton.window, shapes_panes, shapes_label,
            gc, rgc, font_info, BLACK, VERTICAL);
        button = event.xbutton.button;
        inverted_shp = event.xbutton.window;

        while (1)
        {
            while (XCheckTypedEvent(display, ButtonPress, &event));
            XMaskEvent(display, ButtonReleaseMask, &event);
            if (event.xbutton.button == button)
                break;
        }
        if (inverted_shp == event.xbutton.window)
        {
            for (winindex = 0; inverted_shp != shapes_panes[winindex]; winindex++);
            shape_mode = winindex;
        }
        paint_pane(inverted_shp, shapes_panes, shapes_label, gc,
            rgc, font_info, WHITE, VERTICAL);
        inverted_shp = NONE;
        XFlush(display);
    }
    break;
case DestroyNotify:
    break;
case UnmapNotify:
    break;
default:
    ;
}
}
}

```

```

int style_menu(sub_sty, style_panes, style_label)
Window sub_sty;
Window style_panes[];
char *style_label[];
{

```

```

    int winindex;
    int button;
    int i, win;
    XMapSubwindows(display, sub_sty);
    XMapWindow(display, sub_sty);

    while (1)
    {
        XNextEvent(display, &event);
        switch(event.type) {
            case Expose:
                if (inverted_sty == event.xexpose.window)
                    paint_pane(event.xexpose.window, style_panes, style_label,
                        gc, rgc, font_info, BLACK, VERTICAL);
                else
                    paint_pane(event.xexpose.window, style_panes, style_label,
                        gc, rgc, font_info, WHITE, VERTICAL);
                break;
            case ButtonPress:
                if (event.xbutton.window == mainwin) {
                    XUnmapSubwindows(display, sub_sty);
                    XUnmapWindow(display, sub_sty);
                    return (1);
                }
        }
    }
}

```

```

else
if ((event.xbutton.window == panes[0]) ||
    (event.xbutton.window == panes[1]) ||
    (event.xbutton.window == panes[2]) ||
    (event.xbutton.window == panes[3]) ||
    (event.xbutton.window == panes[4]) ||
    (event.xbutton.window == panes[5]) ||
    (event.xbutton.window == panes[6]) ||
    (event.xbutton.window == lin_panes[0]) ||
    (event.xbutton.window == lin_panes[1])) {
XUnmapSubwindows(display, sub_sty);
XUnmapWindow(display, sub_sty);
return (2);
}
else {
paint_pane(event.xbutton.window, style_panes, style_label, gc,
           rgc, font_info, BLACK, VERTICAL);
button = event.xbutton.button;
inverted_sty = event.xbutton.window;

while (1)
{
while (XCheckTypedEvent(display, ButtonPress, &event));
XMaskEvent(display, ButtonReleaseMask, &event);
if (event.xbutton.button == button)
break;
}
if (inverted_sty == event.xbutton.window)
{
win = 0;
while (event.xbutton.window != sty_panes[win])
win +=1;
switch(win) {
case 0:
common_style = LineSolid;
break;
case 1:
common_style = LineOnOffDash;
break;
case 2:
common_style = LineDoubleDash;
break;
}
XSetLineAttributes(display, active_gc, common_width, common_style,
                   cap_style, join_style);
}
paint_pane(inverted_sty, style_panes, style_label, gc, rgc,
           font_info, WHITE, VERTICAL);
inverted_sty = NONE;
XFlush(display);
}
break;

case DestroyNotify:
break;
case UnmapNotify:
break;
default:
;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

width_menu(sub_wth, width_panes, width_label) ถึงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
Window sub_wth;
Window width_panes[];
char *width_label[];

```

int winindex;
int button;
int i,win;
XMapSubwindows(display, sub_wth);
XMapWindow(display, sub_wth);

while (1)
(
    XNextEvent(display, &event);
    switch(event.type) {
    case Expose:
        if (inverted_wth == event.xexpose.window)
            paint_pane(event.xexpose.window, width_panes,width_label,
                gc, rgc, font_info, BLACK, VERTICAL);
        else
            paint_pane(event.xexpose.window, width_panes,width_label,
                gc, rgc, font_info, WHITE, VERTICAL);
        break;
    case ButtonPress:
        if (event.xbutton.window == mainwin) {
            XUnmapSubwindows(display, sub_wth);
            XUnmapWindow(display, sub_wth);
            return;
        }
        else
            if ((event.xbutton.window == panes[0]) ||
                (event.xbutton.window == panes[1]) ||
                (event.xbutton.window == panes[2]) ||
                (event.xbutton.window == panes[3]) ||
                (event.xbutton.window == panes[4]) ||
                (event.xbutton.window == panes[5]) ||
                (event.xbutton.window == panes[6]) ||
                (event.xbutton.window == lin_panes[0]) ||
                (event.xbutton.window == lin_panes[1])) {
                XUnmapSubwindows(display, sub_wth);
                XUnmapWindow(display, sub_wth);
                return;
            }
            else {
                paint_pane(event.xbutton.window, width_panes,width_label,
                    gc, rgc, font_info, BLACK, VERTICAL);
                button = event.xbutton.button;
                inverted_wth = event.xbutton.window;

                while (1)
                {
                    while (XCheckTypedEvent(display, ButtonPress, &event));
                    XMaskEvent(display, ButtonReleaseMask, &event);
                    if (event.xbutton.button == button)
                        break;
                }
                if (inverted_wth == event.xbutton.window)
                {
                    win = 0;
                    while (event.xbutton.window != wth_panes[win])
                        win +=1;
                    switch(win) {
                    case 0:
                        common_width = 1;
                        break;
                    case 1:
                        common_width = 2;
                        break;
                    case 2:
                        common_width = 3;
                        break;
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีก: break; หักดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    XSetLineAttributes(display, active_gc, common_width, common_style,
                       cap_style, join_style);
    }
    paint_pane(inverted_wth, width_panes, width_label, gc, rgc,
               font_info, WHITE, VERTICAL);
    inverted_wth = NONE;
    XFlush(display);
    }
    break;

case DestroyNotify:
    break;
case UnmapNotify:
    break;
default:
    ;
}
}
}

```

```

create_win()
{

```

```

    XCharStruct ovl;
    int winindex, sub_height, sub_width;
    int panes_width, panes_height;
    int dir, asc, des;

```

```

    XTextExtents(font_info, file_label[1], 4, &dir, &asc, &des, &ovl);
    sub_width = ovl.width + 15;
    panes_height = ovl.ascent + ovl.descent + 4;
    sub_height = panes_height * 4 + 28;

```

```

    sub_fi = XCreateSimpleWindow(display, mainwin, 25, 35,
                                sub_width, sub_height, 0, border_pixel,
                                background_pixel);

```

```

    for (winindex = 0; winindex < 4; winindex++)
    {
        fi_panes[winindex] = XCreateSimpleWindow(display, sub_fi, 0,
                                                  sub_height/4*winindex + 3, sub_width - 4,
                                                  panes_height, 2, border_pixel,
                                                  background_pixel);

```

```

        XSelectInput(display, fi_panes[winindex], ButtonPressMask |
                    ButtonReleaseMask | ExposureMask);
    }

```

```

    XTextExtents(font_info, font_label[1], 8, &dir, &asc, &des, &ovl);
    sub_width = ovl.width + 15;
    panes_height = ovl.ascent + ovl.descent + 4;
    sub_height = panes_height * 4 + 28;

```

```

    sub_fnt = XCreateSimpleWindow(display, mainwin, 245, 35,
                                sub_width, sub_height, 0, border_pixel,
                                background_pixel);

```

```

    for (winindex = 0; winindex < 4; winindex++)
    {
        fnt_panes[winindex] = XCreateSimpleWindow(display, sub_fnt, 0,
                                                  sub_height/4*winindex + 3, sub_width - 4,
                                                  panes_height, 2, border_pixel,
                                                  background_pixel);

```

```

        XSelectInput(display, fnt_panes[winindex], ButtonPressMask |
                    ButtonReleaseMask | ExposureMask);
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่ควรเผยแพร่ภายนอก
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ทำซ้ำหรือดัดแปลงในลักษณะใดๆ ที่อาจก่อให้เกิดความเสียหาย
 อย่างถึงแก่กรรมแก่เจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

XTextExtents(font_info, color_label[1], 6, &dir, &asc, &des, &ovl);
sub_width = ovl.width + 15;
panes_height = ovl.ascent + ovl.descent + 4;
sub_height = panes_height * 5 + 28;

sub_clr = XCreateSimpleWindow(display, mainwin, 364,35,
                             sub_width, sub_height,0 ,border_pixel,
                             background_pixel);

for (winindex = 0; winindex < 5; winindex++)
{
    clr_panes[winindex] = XCreateSimpleWindow(display, sub_clr, 0,
        sub_height/5*winindex + 3, sub_width - 4,
        panes_height, 2, border_pixel,
        background_pixel);

    XSelectInput(display, clr_panes[winindex], ButtonPressMask |
        ButtonReleaseMask | ExposureMask);
}

XTextExtents(font_info, line_label[1], 5, &dir, &asc, &des, &ovl);
sub_width = ovl.width + 15;
panes_height = ovl.ascent + ovl.descent + 4;
sub_height = panes_height * 2 + 20;

sub_lin = XCreateSimpleWindow(display, mainwin, 485,35,
                             sub_width, sub_height,0 , border_pixel,
                             background_pixel);

for (winindex = 0; winindex < 2; winindex++)
{
    lin_panes[winindex] = XCreateSimpleWindow(display, sub_lin, 0,
        sub_height/2*winindex + 3, sub_width - 4,
        panes_height, 2, border_pixel,
        background_pixel);

    XSelectInput(display, lin_panes[winindex], ButtonPressMask |
        ButtonReleaseMask | ExposureMask | SubstructureNotifyMask);
}

XTextExtents(font_info, shapes_label[4], 9, &dir, &asc, &des, &ovl);
sub_width = ovl.width + 15;
panes_height = ovl.ascent + ovl.descent + 4;
sub_height = panes_height * 5 + 28;

sub_shp = XCreateSimpleWindow(display, mainwin, 582,35,
                             sub_width, sub_height,0 ,border_pixel,
                             background_pixel);

for (winindex = 0; winindex < 5; winindex++)
{
    shp_panes[winindex] = XCreateSimpleWindow(display, sub_shp, 0,
        sub_height/5*winindex + 3, sub_width - 4,
        panes_height, 2, border_pixel,
        background_pixel);

    XSelectInput(display, shp_panes[winindex], ButtonPressMask |
        ButtonReleaseMask | ExposureMask);
}

XTextExtents(font_info, style_label[1], 5, &dir, &asc, &des, &ovl);
sub_width = ovl.width + 15;
panes_height = ovl.ascent + ovl.descent + 4;
sub_height = panes_height * 3 + 28;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆก็ตาม ขอสงวนสิทธิ์ในเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

sub_sty = XCreateSimpleWindow(display, mainwin, 560,55,
                             sub_width, sub_height,0 ,border_pixel,
                             background_pixel);

for (winindex = 0; winindex < 3; winindex++)
{
    sty_panes[winindex] = XCreateSimpleWindow(display, sub_sty, 0,
        sub_height/3*winindex + 3, sub_width - 4,
        panes_height, 2, border_pixel,
        background_pixel);

    XSelectInput(display, sty_panes[winindex], ButtonPressMask |
        ButtonReleaseMask | ExposureMask);
}

XTextExtents(font_info, width_label[1], 4, &dir, &asc, &des, &ovl);
sub_width = ovl.width + 15;
panes_height = ovl.ascent + ovl.descent + 4;
sub_height = panes_height * 3 + 28;

sub_wth = XCreateSimpleWindow(display, mainwin, 560,75,
                             sub_width, sub_height,0 ,border_pixel,
                             background_pixel);

for (winindex = 0; winindex < 3; winindex++)
{
    wth_panes[winindex] = XCreateSimpleWindow(display, sub_wth, 0,
        sub_height/3*winindex + 3, sub_width - 4,
        panes_height, 2, border_pixel,
        background_pixel);

    XSelectInput(display, wth_panes[winindex], ButtonPressMask |
        ButtonReleaseMask | ExposureMask);
}

load_font(window, f_panes, active_gc)
Window window, f_panes[];
GC active_gc;
{
    int win = 0;

    while (window != f_panes[win])
        win += 1;

    switch(win) {
    case 0:
        txt_buffer[txt_count].fnt = font_label[0];
        font = XLoadFont(display, "8x13");
        common_font = "8x13";
        break;
    case 1:
        txt_buffer[txt_count].fnt = font_label[1];
        font = XLoadFont(display, "8x13bold");
        common_font = "8x13bold";
        break;
    case 2:
        txt_buffer[txt_count].fnt = font_label[2];
        font = XLoadFont(display, "9x15");
        common_font = "9x15";
        break;
    case 3:
        txt_buffer[txt_count].fnt = font_label[3];
        font = XLoadFont(display, "thai24");
        common_font = "thai24";
        fno = 3;
    }
}

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต

เอกสารนี้เป็นการศึกษาเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น เอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    }
    XSetFont (display, active_gc, font);
}

create_gc ()
{
    gc = XCreateGC (display, mainwin, 0, NULL);
    XSetForeground (display, gc, BlackPixel (display, screen));
    rgc = XCreateGC (display, mainwin, 0, NULL);
    XSetForeground (display, rgc, background_pixel);

    valuemask = GCForeground | GCBackground | GCLineStyle;
    values.line_style = LineSolid;
    values.foreground = BlackPixel (display, screen);
    values.background = WhitePixel (display, screen);

    active_gc = XCreateGC (display, mainwin, valuemask, &values);
    XSetFunction (display, active_gc, GXxor);
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

/* THAI.AFB */

StartFont 1.4
Comment Thai Font 1.0
Font thai
Size 24 72 72
FontBoundingBox 18 24 0 -4
Chars 127
Comment encoding unknown
StartProperties
Font_Ascent 20
Font_Descent 4
EndProperties
StartChar <unknown>
Encoding 100
DWidth 18 0
BBX 18 24 0 -4

BitMap

00000

00000

00000

00000

00000

03F00

0C0C0

10060

1C060

02060

03060

06060

0C060

0C060

0C060

0C060

0C060

0C060

0C060

0C060

00000

00000

00000

00000

EndChar

StartChar <unknown>

Encoding 45

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000

00000

00000

00000

00000

07C60

09660

09360

09360

06260

00460

01860

01860

01860

01860

01860

01860



เอกสารนี้เป็นทรัพย์สินส่วนหลวงไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EndChar
StartChar <unknown>
Encoding 39
DWidth 18 0
BBX 18 24 0 -4
BitMap
00000
00000
00000
00000
00000
001C0
00220
00220
00220
001E0
0C060
06060
03060
01860
00860
00460
00660
00360
00160
000C0
00000
00000
00000
00000
EndChar

StartChar <unknown>
Encoding 48
DWidth 18 0
BBX 18 24 0 -4
BitMap
00000
00000
00000
00000
00000
00000
07F00
0C0C0
10040
00060
00060
00C60
01260
01360
01360
00D60
000E0
000E0
00060
00060
00040
00000
00000
00000
00000
EndChar

StartChar <unknown>
Encoding 67
DWidth 18 0
BBX 18 24 0 -4
BitMap



เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของสำนักงานส่งเสริมการศึกษานอกระบบและการศึกษาตามอัธยาศัย (กศน.) ห้ามมิให้คัดลอกหรือเผยแพร่โดยไม่ได้รับอนุญาตจาก กศน. หากฝ่าฝืนจะดำเนินการตามกฎหมายที่เกี่ยวข้อง
ไม่ว่ากรณีใดๆ ห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

39260
25380
25380
18460
00460
01860
01860
01860
01860
01860
01860
01860
01860
01860
01860
01860
07FE0
00000
00000
00000
00000

EndChar
StartChar <unknown>
Encoding 71
DWidth 18 0
BBX 18 24 0 -4

BitMap
00000
00000
00000
00000
00000
0F818
10618
20218
30318
0C318
18318
30318
30318
30318
30318
30318
38718
249D8
24938
24918
18608
00000
00000
00000
00000

EndChar
StartChar <unknown>
Encoding 80
DWidth 18 0
BBX 18 24 0 -4

BitMap
00000
00000
00000
00000
00000
0F818
10618
20218
30318
0C318
10318
30318



เอกสารนี้...
ไม่ว่าการ...
10318
30318

30318
30318
30318
38318
24318
24318
24318
183F8
00000
00388
002D8
001F0
EndChar
StartChar <unknown>
Encoding 69
DWidth 18 0
BBX 18 24 0 -4

BitMap
00000
00000
00000
00000
00000
01F80
06060
08030
0E030
01030
01830
03030
06030
06030
06030
1E030
22030
22030
22230
1C230
01E30
067B0
06670
03C30

EndChar
StartChar <unknown>
Encoding 68
DWidth 18 0
BBX 18 24 0 -4

BitMap
00000
00000
00000
00000
00000
00000
00000
03FC0
04030
07018
00818
00C18
01818
03018
03018
03018
0F018
11018
11018



เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าการพิมพ์ซ้ำ หรือการดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11118
0E318
00618
07F78
0CED8
078D8
EndChar
StartChar <unknown>
Encoding 123
DWidth 18 0
BBX 18 24 0 -4

BitMap
00000
00000
00000
00000
00030
03F30
040C0
08000
13F00
1C0C0
00060
00C60
01230
01230
01330
00D30
00120
00160
00140
00980
1C800
23130
23390
1CC70

EndChar
StartChar <unknown>
Encoding 82
DWidth 18 0
BBX 18 24 0 -4

BitMap
00000
00000
00000
00000
00000
1D860
2A490
2A498
2A918
18918
01218
03218
03418
03418
03818
03018
03018
03018
03018



เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00000
00000

EndChar
StartChar <unknown>
Encoding 60
DWidth 18 0
BBX 18 24 0 -4
BitMap
00000
00000
00000
00000
00000
00000
0CC18
13218
33318
30318
30318
30318
33318
34B18
34B18
33318
31718
129D8
1A938
0C918
0C618
00000
00000
00000
00000
EndChar
StartChar <unknown>
Encoding 73
DWidth 18 0
BBX 18 24 0 -4
BitMap
00000
00000
00000
00000
00000
0F030
10C30
20430
30630
0C630
18630
30630
30630
30630
30630
38648
246C8
24748
24748
18630
00000
00000
00000
00000
EndChar
StartChar <unknown>
Encoding 102
DWidth 18 0
BBX 18 24 0 -4
BitMap



00000
00000
00000
00000
00000
01E00
06180
08040
18060
18060
18C60
19260
19260
19260
18C60
18460
08860
0D060
06060
06060
00000
00000
00000
00000

EndChar
StartChar <unknown>
Encoding 57
DWidth 18 0
BBX 18 24 0 -4

BitMap
00000
00000
00000
00000
00000
00000
07380
08C40
18C60
18060
18060
18C60
19260
19260
19260
18C60
18460
08860
0D060
06060
06060
00000
00000
00000
00000

EndChar
StartChar <unknown>
Encoding 53
DWidth 18 0
BBX 18 24 0 -4

BitMap
00000
00000
00000
00000
00000
03F00



เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าการพิมพ์หรือการอื่นใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0C0C0
10040
1C060
02060
03060
06060
0C060
0C060
0C060
0F060
08860
08860
08860
07060
00000
00000
00000
00000

EndChar
StartChar <unknown>
Encoding 109
DWidth 18 0
BBX 18 24 0 -4
BitMap

00000
00000
00000
00000
00000
0E1C0
11260
11230
11430
0F430
03830
03830
03030
03030
03030
03030
03030
03030
03030
03030
03030
00000
00000
00000
00000

EndChar
StartChar <unknown>
Encoding 84
DWidth 18 0
BBX 18 24 0 -4
BitMap

00000
00000
00000
00000
00000
03E30
0C1B0

18060
33E00
3C380
000C0
060C0



เอกสารนี้เป็นที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าการพิมพ์หรือทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00000
00000
00000
00000
EndChar
StartChar <unknown>
Encoding 120
DWidth 18 0
BBX 18 24 0 -4

BitMap

00030

00030

00030

00030

00030

0E030

11030

11030

11030

0F030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030

03030



EndChar
StartChar <unknown>
Encoding 122
DWidth 18 0
BBX 18 24 0 -4

BitMap

00000

00000

00000

00000

00000

0E060

11060

11060

11060

1E060

18060

18060

18C60

18C60

18C60

18C60

19260

19260

1A160

0C0C0

04080

00000

00000

00000

00000

00000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EndChar
StartChar <unknown>

Encoding 47
DWidth 18 0
BBX 18 24 0 -4
BitMap
00030
00030
00030
00030
00030
07030
08830
08830
08830
0F030
0C030
0C030
0C630
0C630
0C630
0C930
0C930
0D0B0
06060
02040
00000
00000
00000
00000
EndChar
StartChar <unknown>
Encoding 114
DWidth 18 0
BBX 18 24 0 -4
BitMap
00000
00000
00000
00000
00000
00000
1C030
22230
22230
22230
1E230
06230
06230
06530
06530
06530
068B0
068B0
068B0
03060
01040
00000
00000
00000
00000
EndChar
StartChar <unknown>



เอกสารนี้เป็นทรัพย์สินของหอสมุดแห่งชาติสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าการแสวงหาผลประโยชน์อื่นใดอีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
BitMap
00030
00030

11060
0F060
03060
03060
03060
03060
0F060
13C60
13360
131E0
120E0
0E060
00000
00000
00000
00000
EndChar
StartChar <unknown>
Encoding 112
DWidth 18 0
BBX 18 24 0 -4

BitMap

00000
00000
00000
00000
00000
03C60
07260
0D260
0D260
0CC60
06060
03060
00E60
03060
06060
0C060
0C060
0C060
0C060
0FFF0
00000
00000
00000
00000

EndChar

StartChar <unknown>
Encoding 105
DWidth 18 0
BBX 18 24 0 -4

BitMap

00000
00000
00000
00000
00000
00000
01F30
060C0
0C000
0BF00
0C180
000C0
000C0
000C0
000C0



เอกสารนี้เป็นที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000C0
003C0
00440
00440
00440
00380
00000
00000
00000
00000
EndChar
StartChar <unknown>
Encoding 65
DWidth 18 0
BBX 18 24 0 -4

BitMap
00000
00000
00000
00000
00000
00000
01F00
060C0
08040
0E060
01060
01860
03060
06060
06060
06060
07860
04460
04460
04460
03860
00060
00060
00060
00060
00060

EndChar
StartChar <unknown>
Encoding 93
DWidth 18 0
BBX 18 24 0 -4

BitMap
00000
00000
00000
00000
00000
00000
03F00
0C180
180C0
180C0
00060
01860
03460
06260
06260
0C160
0F160
088E0
088E0
08860
07060



เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าการอื่นใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00000
00000
00000
EndChar
StartChar <unknown>
Encoding 49
DWidth 18 0
BBX 18 24 0 -4

BitMap

00000

00000

00000

00000

00000

03F00

0C180

180C0

180C0

00060

01860

03460

06260

06260

0C160

0F160

088E0

088E0

08860

07060

00000

00000

00000

00000

EndChar

StartChar <unknown>

Encoding 59

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000

00000

00000

00000

00000

01F00

060C0

0C040

0C060

00060

00060

00060

00060

00060

00060

001E0

00220

00220

00220

001C0

00000

00000

00000

EndChar

StartChar <unknown>

Encoding 76

DWidth 18 0



เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าการอื่น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BBX 18 24 0 -4

BitMap

00000

00018

00018

00018

00030

03F60

0C0C0

081C0

18060

18060

18C60

19260

19260

1B260

0AC60

0A060

0E060

06060

06060

06060

00000

00000

00000

00000

EndChar

StartChar <unknown>

Encoding 75

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000

00000

00000

00000

00000

1C0C0

220C0

220C0

220C0

1E0C0

066C0

06958

06958

067D0

060E0

060C0

060C0

060C0

060C0

1FFC0

00000

00000

00000

00000

EndChar

StartChar <unknown>

Encoding 108

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000

00008

00008

00008



เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าการคัดลอกหรือการนำออกนอกระบบอื่น ๆ ก็ห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00030
03F60
0C1C0
181C0
18060
00060
01860
03460
06260
06260
0C160
0F160
088E0
088E0
08860
07060
00000
00000
00000
00000

EndChar

StartChar <unknown>

Encoding 115

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000

00000

00000

00000

00000

1C0E0

22110

22110

22110

1C0E0

060B0

06130

06230

06230

06430

06830

07030

07030

06030

06030

00000

00000

00000

00000

EndChar

StartChar <unknown>

Encoding 62

DWidth 18 0

BBX 18 24 0 -4

BitMap

00798

018D8

01860

01860

007B0

1C030

22030

22030

22030

1E030



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าการคัดลอกหรือการนำออกนอกระบบอื่น ๆ ก็ห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

06030
06230
06230
06230
06530
06530
068B0
068B0
03060
01040
00000
00000
00000
00000

EndChar

StartChar <unknown>

Encoding 118

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000
00000
00000
00000
00000
03F00
060C0
0C040
00060
00060
03860
04460
04460
04460
07860
06060
06060
06060
06060
0FFE0
00000
00000
00000
00000

EndChar

StartChar <unknown>

Encoding 86

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000
00030
00030
03C60
0C3C0
0C180
0C380
03CC0
000C0
000C0
070C0



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าการ
088C0
088C0
088C0
0F0C0
080C0
080C0

080C0
080C0
1FFC0
00000
00000
00000
00000
EndChar
StartChar <unknown>
Encoding 113
DWidth 18 0
BBX 18 24 0 -4

BitMap

00000

00000

00000

00000

00000

0E060

11070

110F0

121B0

0C330

03C30

00030

00030

00030

00030

00030

00030

00030

00000

00000

00000

00000

EndChar

StartChar <unknown>

Encoding 116

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000

00000

00000

00000

00000

00000

00000

0E060

11060

110C0

1E180

0FE00

00000

00000

0E060

11060

110C0

1E180

0FE00

00000

00000

00000

00000

EndChar



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

StartChar <unknown>

Encoding 121

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

03818

04418

04418

07830

03FC0

EndChar

StartChar <unknown>

Encoding 107

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000

00000

00000

00000

00000

01F00

060C0

0C040

0C060

00060

00060

00060

00060

00060

00060

00060

00060

00060

00060

00000

00000

00000

00000

EndChar

StartChar <unknown>

Encoding 46

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000



เอกสารนี้เป็นทรัพย์สินส่วนราชการใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าการใด ๆ ทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BitMap

00000

07E30

181C0

30000

3FC00

00600

00300

00300

00300

00300

00300

00300

00300

00300

00300

003C0

00220

00220

001C0

00000

00000

00000

00000

EndChar

StartChar <unknown>

Encoding 31

DWidth 18 0

BBX 18 24 0 -4

BitMap

07E00

18180

330C0

338C0

318C0

1F0C0

06180

00180

00300

00300

00300

00300

00300

00300

00300

003C0

00220

00220

001C0

00000

00000

00000

00000

EndChar

StartChar <unknown>

Encoding 119

DWidth 18 0

BBX 18 24 0 -4

BitMap

200C0

301C0

182C0

0C4C0

068C0



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00060
00060
000C0
01F80
06000
0C000
0C060
0CC90
0CC90
073E0
00000
00000
EndChar
StartChar <unknown>
Encoding 106
DWidth 18 0
BBX 18 24 0 -4

BitMap
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00060
00060
00060
00060
00060
00060
00000
EndChar

StartChar <unknown>
Encoding 104
DWidth 18 0
BBX 18 24 0 -4

BitMap
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000

00000
00000
03818
04418
04418



เอกสารนี้เป็นทรัพย์สินส่วนราชการที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00000
00000
00000
00000
00000
00000
00000
00000
0FF00
1C180
380C0
30060
31860
31860
39860
1F060
000C0
00180
07F00
00000
00000
00000
00000

EndChar

StartChar <unknown>

Encoding 35

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

60000

61DC0

33260

33260

33260

1B060

1B460

19C60

0C060

0C060

0FFE0

00000

00000

00000

00000

EndChar

StartChar <unknown>

Encoding 36

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000

00000

00000

00000

00000

00000

00000



เอกสารนี้เป็นที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00000
0E380
194C0
31C60
30860
30860
30860
36060
33060
33060
33060
1E060
00000
00000
00000
00000
EndChar
StartChar <unknown>
Encoding 37
DWidth 18 0
BBX 18 24 0 -4

BitMap
00000
00000
00000
00000
00000
00000
000C0
000C0
000C0
00180
07F00
0C000
0C000
181C0
18260
18660
187C0
18700
18300
18180
0FFC0
00000
00000
00000
00000

EndChar
StartChar <unknown>
Encoding 42
DWidth 18 0
BBX 18 24 0 -4

BitMap
00000
00000
00000
00198
00258
00270
001E0
00060
000C0
07F80
0C000
0C000
181C0
18260



เอกสารนี้ได้รับการสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18260
183C0
18300
18300
18180
0FFEO
00000
00000
00000
00000
EndChar
StartChar <unknown>
Encoding 40
DWidth 18 0
BBX 18 24 0 -4

BitMap

00000
00000
00000
00000
00000
00000
60000
30000
18000
0C000
0DF80
060C0
00060
03860
06460
06460
06460
06860
060E0
030C0
01F80
00000
00000
00000
00000

EndChar

StartChar <unknown>
Encoding 41
DWidth 18 0
BBX 18 24 0 -4

BitMap

00000
00000
00000
00000
00000
00030
00030
00030
00030
00030
0F730
19D30
308B0
308B0
308B0
348F0
320F0
320E0
320E0
320C0
1C0C0



เอกสารนี้เป็นทรัพย์สินส่วนพระมหากษัตริย์ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00000
00000
00000
00000
EndChar
StartChar <unknown>
Encoding 95
DWidth 18 0
BBX 18 24 0 -4
BitMap
00000
00000
00000
00000
00060
00060
000C0
00180
07F00
0C000
1C000
181E0
18130
18130
18830
18C60
19660
0D2C0
061C0
00000
00000
00000
00000
EndChar
StartChar <unknown>
Encoding 43
DWidth 18 0
BBX 18 24 0 -4
BitMap
00000
00000
00000
00000
00000
00010
00030
00360
004C0
00400
07800
18C00
18600
18600
1A300
19300
19000
0D000
0F000
00000
00000
00000



เอกสารนี้เป็นทรัพย์สินที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
EndChar
StartChar <unknown>
Encoding 90

DWidth 18 0
BBX 18 24 0 -4

BitMap

00000

00000

00000

00100

00200

00400

00400

00800

01800

01800

01800

01800

01800

01800

00800

00400

00400

00200

00100

00000

00000

00000

00000

00000

EndChar

StartChar <unknown>

Encoding 88

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000

00000

00000

02000

01000

00800

00800

00400

00600

00600

00600

00600

00600

00600

00400

00800

00800

01000

02000

00000

00000

00000

00000

EndChar

StartChar <unknown>

Encoding 89

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000

00000

00000



เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าการอื่นใดที่ขัดแย้งกับลิขสิทธิ์อื่น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00000
00000
00C00
00C00
00C00
00C00
00C00
00C00
00C00
00C00
1FFE0
00C00
00C00
00C00
00C00
00C00
00C00
00000
00000
00000
00000
00000

EndChar
StartChar <unknown>
Encoding 125
DWidth 18 0
BBX 18 24 0 -4
BitMap

00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
03C00
07E00
06600
03E00
00600
00C00
01800
03000

EndChar
StartChar <unknown>
Encoding 51
DWidth 18 0
BBX 18 24 0 -4
BitMap

00000
00000
00000
00000
00000
00000
00000
00000



เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00000
00000
00000
1FFEO
1FFEO.
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000

EndChar

StartChar <unknown>

Encoding 34

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
01C00
03E00
03E00
01C00
00000
00000
00000
00000

EndChar

StartChar <unknown>

Encoding 50

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000
00000
00000
000C0
000C0
00180
00180
00300
00300
00600
00600
000C0
000C0
01800
01800



เอกสารนี้เป็นทรัพย์สินส่วนราชการที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

03000
03000
06000
06000
0C000
00000
00000
00000
00000
EndChar
StartChar <unknown>
Encoding 87
DWidth 18 0
BBX 18 24 0 -4

BitMap

02080

071C0

071C0

071C0

071C0

02080

02080

02080

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

00000

EndChar

StartChar <unknown>

Encoding 66

DWidth 18 0

BBX 18 24 0 -4

BitMap

00000

00000

00000

000C0

000C0

1E180

21180

21300

21300

1E600

00600

00C00

00C00

019E0

01A10

03210

03210

061E0

06000

0C000

00000



เอกสารนี้เป็นทรัพย์สินที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

06000

0C000

00000

00000
00000
00000
EndChar
StartChar <unknown>
Encoding 63
DWidth 18 0
BBX 18 24 0 -4

BitMap

00000
00000
00000
07F80
0C060
18030
18030
0C030
00060
000C0
00180
00300
00600
00600
00C00
00C00
00C00
00000
00C00
01E00
00C00
00000
00000
00000

EndChar
StartChar <unknown>
Encoding 77
DWidth 18 0
BBX 18 24 0 -4

BitMap

00000
00000
00000
07F80
0C060
18030
18030
0C030
00060
000C0
00180
00300
00600
00600
00C00
00C00
00C00
00000
00C00
01E00
00C00
00000



เอกสารนี้เป็นทรัพย์สินของหอสมุดแห่งชาติสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ หากมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อเจ้าหน้าที่หอสมุดแห่งชาติ โทร. 0-2611-1111
EndChar
StartChar <unknown>
Encoding 101

DWidth 18 0
BBX 18 24 0 -4
BitMap
00000
30000
48000
48000
30000
01F00
060C0
0C040
0C060
00060
00060
00060
00060
00060
00060
00060
00060
00060
00060
00000
00000
00000
00000

EndChar
StartChar <unknown>
Encoding 38

DWidth 18 0
BBX 18 24 0 -4

BitMap
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000
00000

03818
04418
04418
03C18
00C300
01860
07F80
00000
03818
04418
04418
07830
03FC0

EndChar
StartChar <unknown>
Encoding 0

DWidth 18 0
BBX 18 24 0 -4

BitMap
00000
00000
00000



เอกสารนี้เป็นทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ เว้นแต่ได้รับอนุญาตจากมหาวิทยาลัยฯ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

