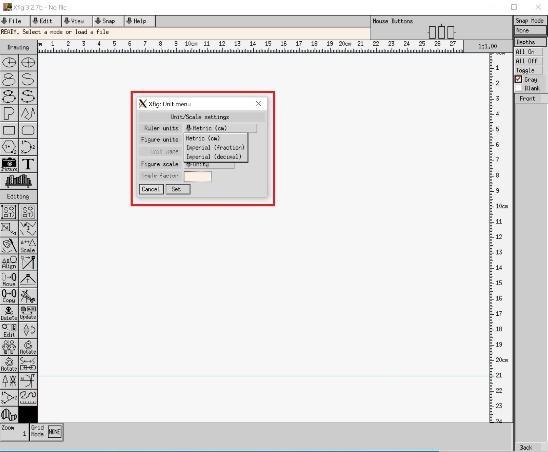
**Software Change Request**

| **Software** | **Baseline Version** | **Feature Name** | **Difficulty** |
| --- | --- | --- | --- |
| Xfig | 3.2.8a | Shortcut Button for Toggling Unit of Length | Easy (Est. 5 files; 30 LOC) |

***Current Behaviors:***

Currently the default unit of length is Metric (centimeter, or cm), which can be changed to Imperial through the menu (Menu -> Edit -> Set Units), which is shown in Figure 1. However, Xfig users always prefer doing actions through shortcut buttons on the Editing Mode Panel on the left hand side of Xfig.

*Figure 1: the unit setting window of Xfig*

***Expected Behavior:***

Create a unit toggling button on the Editing Mode Panel on the left hand side of Xfig. This toggling button should deliver the same behavior as the menu version: First click will change the unit to Imperial, and one more clicking will change it back to Metric.

***Solution Hints for Instructor:***

This project can be broken into two parts: adding the function, and adding the button. The function needs to change the internal units, then update the interface. This is most appropriately created in the w\_rulers.c file, which naturally means w\_rulers.h will need to be modified as well. Adding the button involves modifying w\_modepanel.c by adding an entry to the mode\_switches array. This is described in the undo button solution, which explains that:

“The entry that needs to be added is a mode\_sw\_info struct, which as the format:

icon\_struct \*icon; //pointer to the icon of the button  
int mode; // mode xfig enters, such as creating a box or moving an object  
void (\*setmode\_func) (); // the function that executes when the button is pressed  
int objmask; //objects that will be affected  
unsigned long indmask; // mask to display indicators  
char modemsg[MAX\_MODEMSG\_LEN]; // message for function  
Boolean popup; // if something will popup  
Widget widget; // widget to open  
Pixmap pixmap, reversePM; // pixmaps forXtVaCreateManagedWidget()

All buttons use the same values for popup, widget, and the pixmaps, with the rest of the values being self-explanatory. setmode\_func will be a pointer to the undo function, so u\_undo.h will need to be included.”

In this case the function will be pointing to a toggle\_units function, so for the same reason as above, the w\_rulers.c file must be included. Adding an icon for a button is also explained in the undo button project solution:

“All of xfig’s button icons are stored in w\_icons.c, with each button having a normal and small icon. These icons are stored in a bitmap format, which can easily be obtained from an image manipulation program. The undo icon struct, as well as the length and width of both icons, must be set here. In w\_icons.h, the icon must be defined as an external variable.”

**w\_rulers.c:**

… **anywhere**

+ void toggle\_in\_cm(void)  
+ {  
+ appres.INCHES = !appres.INCHES; // switch between inches and not inches (cm)  
+ cur\_gridunit = appres.INCHES; // switch between decimal and imperial units  
+   
+ set\_unit\_indicator(False); // Make scale unit panel update to according units  
+  
+ //Update units of every drawn object (taken from unit\_panel\_set())  
+ if (!emptyfigure()) {  
+ if (!appres.INCHES)  
+ read\_scale\_compound(&objects,(2.54\*PPCM)/((float)PPI),0);  
+ else  
+ read\_scale\_compound(&objects,((float)PPI)/(2.54\*PPCM),0);  
+ }  
+  
+ redisplay\_canvas(); // Draw everything with changed values  
+  
+}

**w\_rulers.h:**

… **anywhere**

+ extern void toggle\_in\_cm(void);

**w\_modepanel.c:**

… **includes**

#include "w\_mousefun.h"  
#include "w\_msgpanel.h"  
+ #include “w\_rulers.h”  
#include "w\_setup.h"  
#include "w\_util.h"

.. end of **mode\_sw\_info mode\_switches[]**

{&areameas\_ic, F\_AREAMEAS, areameas\_selected, M\_AREAMEAS\_OBJECT, I\_MIN2,  
 "Measure AREA of polygons, arcs and ellipses (Ctrl-m)",  
 False, NULL, (Pixmap)0, (Pixmap)0},  
+ {&unittoggle\_ic, F\_NULL, toggle\_in\_cm, M\_ALL, I\_NONE,  
+ "Toggle units between inches and centimeters",  
+ False, NULL, (Pixmap)0, (Pixmap)0},

**w\_icons.c:**

… “**icons for mode panel” section**

#define areameas\_width\_small 22  
#define areameas\_height\_small 22  
static unsigned char areameas\_bits\_small[] = {  
0x00,0x00,0xc0,0x00,0x00,0xc0,0x00,0x00,0xc0,0x80,0x01,0xc0,0xc0,0x03,0xc0,  
0x60,0x05,0xc0,0x50,0x0d,0xc0,0x58,0x15,0xc0,0x54,0x15,0xc0,0x56,0x35,0xc0  
0x56,0x35,0xcc,0x56,0xf5,0xd7,0x56,0x55,0xd5,0x56,0x55,0xd5,0xdc,0x57,0xd5,  
0x78,0x5c,0xcd,0x18,0x58,0xc7,0x00,0xf8,0xc1,0x00,0x20,0xc0,0x00,0x00,0xc0,  
0x00,0x00,0xc0,0x00,0x00,0xc0};  
+ #define unittoggle\_width\_small 22  
+ #define unittoggle\_height\_small 22  
+ static unsigned char unittoggle\_bits\_small[] = { // example toggle icon  
+ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00,  
+ 0x80, 0x01, 0x00, 0x48, 0x02, 0x18, 0x48, 0x02, 0x06, 0x48, 0x82, 0x01,  
+ 0x48, 0x42, 0x00, 0x00, 0x30, 0x00, 0x00, 0x0c, 0x00, 0x00, 0x03, 0x00,  
+ 0xc0, 0x00, 0x00, 0x30, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x86, 0x63, 0x03,  
+ 0x40, 0x90, 0x04, 0x40, 0x90, 0x04, 0x40, 0x90, 0x04, 0x80, 0x93, 0x04,  
+ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };

… “**NORMAL SIZE ICONS” section**

#define lenmeas\_width\_big 36  
#define lenmeas\_height\_big 32  
static unsigned char lenmeas\_bits\_big[] = {  
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
0x00,0x00,0x00,0x00,0x1e,0x00,0x00,0x00,0x80,0x13,0x00,0x40,  
0x00,0x70,0x10,0x80,0xc1,0x00,0x18,0x10,0xe0,0x83,0x01,0x08,  
0x10,0x30,0x06,0x01,0x0c,0x10,0x10,0x0c,0x01,0x04,0x18,0x18,  
0x08,0x01,0x04,0x08,0x08,0x0c,0x01,0x04,0x04,0x08,0x04,0x01,  
0x00,0x03,0x0c,0x84,0x01,0xc0,0x00,0x06,0xdc,0x00,0x60,0x00,  
0x03,0x70,0x00,0x60,0xc0,0x00,0x00,0x00,0xc0,0x3f,0x00,0x00,  
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x08,0x21,0x84,0x10,0x02,  
0x08,0x21,0x84,0x10,0x02,0x08,0x21,0x84,0x10,0x02,0xf8,0xff,  
0xff,0xff,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
0x00,0x00,0x00,0x00};

+ #define unittoggle\_width\_big 36  
+ #define unittoggle\_height\_big 32  
+ static unsigned char unittoggle\_bits\_big[] = { //example toggle icon  
+ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
+ 0x00, 0x60, 0x00, 0x00, 0x00, 0x00, 0x70, 0x00, 0x00, 0x00, 0x00, 0x38,  
+ 0x00, 0x40, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x38, 0x00, 0x0e, 0x00, 0x40,  
+ 0x44, 0x00, 0x07, 0x00, 0x40, 0x44, 0x80, 0x03, 0x00, 0x40, 0x44, 0xc0,  
+ 0x01, 0x00, 0x40, 0x44, 0xe0, 0x00, 0x00, 0x40, 0x44, 0x70, 0x00, 0x00,  
+ 0x40, 0x44, 0x38, 0x00, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00, 0x00,  
+ 0x0e, 0x00, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00, 0x80, 0x03, 0x00,  
+ 0x00, 0x00, 0xc0, 0x01, 0x00, 0x00, 0x00, 0xe0, 0x00, 0x00, 0x00, 0x00,  
+ 0x70, 0x00, 0x00, 0x00, 0x00, 0x38, 0x60, 0xdc, 0x01, 0x00, 0x1c, 0x10,  
+ 0x22, 0x02, 0x00, 0x0e, 0x10, 0x22, 0x02, 0x00, 0x07, 0x10, 0x22, 0x02,  
+ 0x80, 0x03, 0x10, 0x22, 0x02, 0xc0, 0x01, 0x60, 0x22, 0x02, 0xe0, 0x00,  
+ 0x00, 0x00, 0x00, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
+ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
+ 0x00, 0x00, 0x00, 0x00 };

… **defining structs section**

icon\_struct areameas\_ic;  
+ icon\_struct unittoggle\_ic;

… **populate\_icons\_big**

icon\_struct lenmeas\_ic\_ = { lenmeas\_width\_big, lenmeas\_height\_big, (char\*)lenmeas\_bits\_big };  
 icon\_struct areameas\_ic\_ = { areameas\_width\_big, areameas\_height\_big, (char\*)areameas\_bits\_big };  
+ icon\_struct unittoggle\_ic\_ = { unittoggle\_width\_big, unittoggle\_height\_big, (char\*)unittoggle\_bits\_big };  
 regpoly\_ic = regpoly\_ic\_;  
 addpt\_ic = addpt\_ic\_;

… **populate\_icons\_big (near bottom)**

anglemeas\_ic = anglemeas\_ic\_;  
 lenmeas\_ic = lenmeas\_ic\_;  
 areameas\_ic = areameas\_ic\_;  
+ unittoggle\_ic = unittoggle\_ic\_;  
}

… **populate\_icons\_small**

icon\_struct lenmeas\_ic\_ = { lenmeas\_width\_small, lenmeas\_height\_small, (char\*)lenmeas\_bits\_small };  
 icon\_struct areameas\_ic\_ = { areameas\_width\_small, areameas\_height\_small, (char\*)areameas\_bits\_small };  
+ icon\_struct unittoggle\_ic\_ = { unittoggle\_width\_small, unittoggle\_height\_small, (char\*)unittoggle\_bits\_small };  
 regpoly\_ic = regpoly\_ic\_;  
 addpt\_ic = addpt\_ic\_;

..**populate\_icons\_small**

lenmeas\_ic = lenmeas\_ic\_;  
 areameas\_ic = areameas\_ic\_;  
+ unittoggle\_ic = unittoggle\_ic\_;  
}

**w\_icons.h:**

extern icon\_struct areameas\_ic;  
+ extern icon\_struct unittoggle\_ic;