**Software Change Request**

| **Software** | **Baseline Version** | **Feature Name** | **Difficulty** |
| --- | --- | --- | --- |
| Xfig | 3.2.8a | Auto Increment Default Depth Value | Easy (Est. 9 files; 33 LOC) |

***Current Behaviors:***

Currently, each Xfig object (e.g., arc, line, rectangle) has a property called depth, which is used to identify and group the objects on the same layer. In Xfig, the valid range of depth is 0 to 999, and the default value is 50 for any newly created object. Also, it means that all objects created are on the same layer by default. Figure 1 shows the default depth value is 50 for the three objects on canvas. A picture containing graphical user interface

Description automatically generated

*Figure 1: the default depth for all objects on canvas*

***Expected Behavior:***

This task requests to change the way of assigning the default depth value to newly created. The default depth will be increased with a certain interval (e.g., 1) for new created objects accordingly. For example, the default depth for the 1st object created on the canvas is 50, and then the 2nd one will be 51, and so on. Also, it means that all objects created are on different layers by default.

***Solution Hints for Instructor:***

This feature may not be justified to be desirable in practice, but it can be used as an exercise for students.

The **cur\_depth** global variable controls the depth at which an object is created. By incrementing this variable whenever an object is added, the depth of the next object will be 1 more than the last. However, this does not immediately update the depth button’s display value. To update this, **show\_depth(depth\_button)** must be called. Consequently, **w\_indpanel** must be included in all modified files.

This solution will be the same in all modified files, with the exception of d\_ellipse.c. Since d\_ellipse.c contains multiple create functions, the latter two lines will be included for the create\_ellipsebyrad, create\_ellipsebydia, create\_circlebyrad, and create\_circlebydia functions. For all other files, code only needs to be added to their includes section, and their respective create\_[object name] functions.

The files that should be modified are:

d\_arc.c d\_arcbox.c d\_box.c d\_ellipse.c d\_line.c  
d\_picobj.c d\_regpoly.c d\_spline.c d\_text.c

**d\_arc.c:**

… **includes**

#include "w\_cursor.h"  
#include "w\_drawprim.h"  
+ #include "w\_indpanel.h"  
#include "w\_msgpanel.h"  
#include "w\_mousefun.h"

… **create\_arcobject(int lx, int ly)**

arc->fill\_color = cur\_fillcolor;  
arc->cap\_style = cur\_capstyle;  
- arc->depth = cur\_depth++;  
+ arc->depth = cur\_depth++;  
+ show\_depth(depth\_button);   
arc->direction = compute\_direction(point[0], point[1], point[2]);  
/\* only allow arrowheads for open arc \*/

**d\_arcbox.c:**

… **includes**

#include "w\_canvas.h"  
#include "w\_cursor.h"  
+ #include "w\_indpanel.h"  
#include "w\_msgpanel.h"  
#include "w\_mousefun.h"

… **create\_arc\_boxobject(int x, int y)**

box->pen\_color = cur\_pencolor;  
box->fill\_color = cur\_fillcolor;  
- box->depth = cur\_depth;  
+ box->depth = cur\_depth++;  
+ show\_depth(depth\_button);  
box->pen\_style = -1;  
box->join\_style = cur\_joinstyle;

**d\_box.c:**

… i**ncludes**

#include "w\_canvas.h"  
#include "w\_cursor.h"  
+ #include "w\_indpanel.h"  
#include "w\_mousefun.h"  
#include "w\_msgpanel.h"

… **create\_boxobject(int x, int y)**

box->pen\_color = cur\_pencolor;  
box->fill\_color = cur\_fillcolor;  
- box->depth = cur\_depth;  
+ box->depth = cur\_depth++;  
+ show\_depth(depth\_button);  
box->pen\_style = -1;  
box->join\_style = cur\_joinstyle;

**d\_ellipse.c:**

… **includes**

#include "w\_canvas.h"  
#include "w\_cursor.h"  
+ #include "w\_indpanel.h"  
#include "w\_mousefun.h"  
#include "xfig\_math.h"

… **create\_ellipsebyrad(int x, int y)**

ellipse->pen\_color = cur\_pencolor;  
ellipse->fill\_color = cur\_fillcolor;  
- ellipse->depth = cur\_depth;  
+ ellipse->depth = cur\_depth++;  
+ show\_depth(depth\_button);  
ellipse->pen\_style = -1;  
ellipse->fill\_style = cur\_fillstyle;

… **create\_ellipsebydia(int x, int y)**

ellipse->pen\_color = cur\_pencolor;  
ellipse->fill\_color = cur\_fillcolor;  
- ellipse->depth = cur\_depth;  
+ ellipse->depth = cur\_depth++;  
+ show\_depth(depth\_button);  
ellipse->pen\_style = -1;  
ellipse->fill\_style = cur\_fillstyle;

… **create\_circlebyrad(int x, int y)**

c->pen\_color = cur\_pencolor;  
c->fill\_color = cur\_fillcolor;  
- c->depth = cur\_depth;  
+ c->depth = cur\_depth++;  
+ show\_depth(depth\_button);  
c->pen\_style = -1;  
c->fill\_style = cur\_fillstyle;

...**create\_circlebydia(int x, int y)**

c->pen\_color = cur\_pencolor;  
c->fill\_color = cur\_fillcolor;  
- c->depth = cur\_depth;  
+ c->depth = cur\_depth++;  
+ show\_depth(depth\_button);  
c->pen\_style = -1;  
c->fill\_style = cur\_fillstyle;

**d\_line.c:**

… **includes**

#include "w\_cursor.h"  
#include "w\_drawprim.h"  
+ #include "w\_indpanel.h"  
#include "w\_mousefun.h"  
#include "w\_msgpanel.h"

… **create\_lineobject(int x, int y)**

line->thickness = cur\_linewidth;  
line->pen\_color = cur\_pencolor;  
line->fill\_color = cur\_fillcolor;  
- line->depth = cur\_depth;  
+ line->depth = cur\_depth++;  
+ show\_depth(depth\_button);  
line->pen\_style = -1;  
line->join\_style = cur\_joinstyle;

**d\_picobj.c**

… **includes**

#include "w\_canvas.h"  
#include "w\_cursor.h"  
+ #include "w\_indpanel.h"  
#include "w\_mousefun.h"  
#include "w\_msgpanel.h"

… **create\_picobj(int x, int y)**

box->pen\_color = cur\_pencolor;  
box->fill\_color = DEFAULT;  
- box->depth = cur\_depth;  
+ box->depth = cur\_depth++;  
+ show\_depth(depth\_button);  
box->pen\_style = -1;  
box->join\_style = 0; /\* not used \*/

d\_regpoly.c

… **includes**

#include "w\_canvas.h"  
#include "w\_cursor.h"  
+ #include "w\_indpanel.h"  
#include "w\_mousefun.h"  
#include "w\_msgpanel.h"

… **create\_regpoly(int x, int y)**

poly->pen\_color = cur\_pencolor;  
poly->fill\_color = cur\_fillcolor;  
- poly->depth = cur\_depth;  
+ poly->depth = cur\_depth++;  
+ show\_depth(depth\_button);  
poly->pen\_style = -1;  
poly->join\_style = cur\_joinstyle;

d\_spline.c

… **includes**

#include "w\_canvas.h"  
#include "w\_cursor.h"  
+ #include "w\_indpanel.h"  
#include "w\_mousefun.h"  
#include "w\_msgpanel.h"

… **create\_splineobject(int x, int y)**

spline->fill\_color = cur\_fillcolor;  
spline->cap\_style = cur\_capstyle;  
- spline->depth = cur\_depth;  
+ spline->depth = cur\_depth++;  
+ show\_depth(depth\_button);  
spline->pen\_style = -1;  
spline->fill\_style = cur\_fillstyle;

d\_text.c

… **includes**

#include "w\_canvas.h"  
#include "w\_drawprim.h"  
+ #include "w\_indpanel.h"  
#include "w\_mousefun.h"  
#include "w\_msgpanel.h"

… **create\_textobject(void)**

new\_t->length = size.length;  
 cur\_t = new\_t;  
}  
+ cur\_depth++;  
+ show\_depth(depth\_button);  
/\* draw it and any objects that are on top \*/  
redisplay\_text(cur\_t);  
}