```c
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <X11/Xlib.h>

#include "flame.h"

//
int flame_flush_display(flame_obj_t *fo)
{
  if (fo && fo->display)
    return XFlush(fo->display), 1;
  else
    return 0;
}

//
int flame_close(flame_obj_t *fo)
{
  if (fo)
  {
    XFreeGC(fo->display, fo->gc);
    XCloseDisplay(fo->display);

    free(fo);

    return 1;
  }
  else
    return 0;
}

//
flame_obj_t *flame_open(char *title, int width, int height)
{
  XEvent e;
  int vb = 0;
  Visual *visual;
  int blackColor, whiteColor;
  XSetWindowAttributes attrib;
  flame_obj_t *fo = malloc(sizeof(flame_obj_t));

  fo->display = XOpenDisplay(0);

  if (!fo->display)
    {
      fprintf(stderr,"flame_open: unable to open the graphics window.\n");
      exit(1);
    }

  visual = DefaultVisual(fo->display, 0);

  fo->fast_color_mode = (visual && visual->class == TrueColor);

  blackColor = BlackPixel(fo->display, DefaultScreen(fo->display));
  whiteColor = WhitePixel(fo->display, DefaultScreen(fo->display));

```

```
58    //fo->window = XCreateSimpleWindow(fo->display, DefaultRootWindow(fo->display),
      0, 0, width, height, 0, blackColor, blackColor);
59    fo->window = XCreateSimpleWindow(fo->display, DefaultRootWindow(fo->display), 0,
      0, width, height, 0, blackColor, blackColor);
60
61    attrib.backing_store = Always;
62
63    XChangeWindowAttributes(fo->display, fo->window, CWBackingStore, &attrib);
64
65    XStoreName(fo->display, fo->window,title);
66
67    XSelectInput(fo->display, fo->window, StructureNotifyMask | ExposureMask |
      KeyPressMask | ButtonPressMask | ButtonReleaseMask | PointerMotionMask);
68
69    XMapWindow(fo->display, fo->window);
70
71    fo->gc = XCreateGC(fo->display, fo->window, 0, 0);
72
73    fo->colormap = DefaultColormap(fo->display, 0);
74
75    XSetForeground(fo->display, fo->gc, whiteColor);
76
77    while(!vb)
78      {
79        XNextEvent(fo->display, &e);
80
81        vb = (e.type == MapNotify);
82      }
83
84    return fo;
85  }
86
87  //
88  void flame_draw_point(flame_obj_t *fo, int x, int y)
89  { XDrawPoint(fo->display, fo->window, fo->gc, x, y); }
90
91  //
92  void flame_draw_line(flame_obj_t *fo, int x1, int y1, int x2, int y2)
93  { XDrawLine(fo->display,fo->window,fo->gc, x1, y1, x2, y2); }
94
95  //
96  void flame_set_color(flame_obj_t *fo, int r, int g, int b)
97  {
98    XColor color;
99
100   if (fo->fast_color_mode)
101     color.pixel = ((b & 0xff) | ((g & 0xff) << 8) | ((r & 0xff) << 16));
102   else
103     {
104       color.pixel = 0;
105       color.red   = r << 8;
106       color.green = g << 8;
107       color.blue  = b << 8;
108       XAllocColor(fo->display, fo->colormap, &color);
109     }
110
111   XSetForeground(fo->display, fo->gc, color.pixel);
```

```
112  }
113
114
115  //
116  void flame_clear_display(flame_obj_t *fo)
117  { XClearWindow(fo->display, fo->window); }
118
119  //
120  void flame_clear_color(flame_obj_t *fo, int r, int g, int b )
121  {
122    XColor color;
123    XSetWindowAttributes attrib;
124
125    color.pixel = 0;
126    color.red   = r << 8;
127    color.green = g << 8;
128    color.blue  = b << 8;
129    XAllocColor(fo->display, fo->colormap, &color);
130
131    attrib.background_pixel = color.pixel;
132    XChangeWindowAttributes(fo->display, fo->window, CWBackPixel,&attrib);
133  }
134
135  //
136  int flame_event_waiting(flame_obj_t *fo)
137  {
138    XEvent event;
139
140    flame_flush_display(fo);
141
142    while (1)
143      {
144        if (XCheckMaskEvent(fo->display, -1, &event))
145      {
146        if (event.type == KeyPress)
147          {
148            XPutBackEvent(fo->display, &event);
149
150            return 1;
151          }
152        else
153          if (event.type == ButtonPress)
154            {
155          XPutBackEvent(fo->display, &event);
156
157          return 1;
158            }
159          else
160            return 0;
161      }
162        else
163      return 0;
164      }
165  }
166
167  //
168  char flame_wait(flame_obj_t *fo, int *click_x, int *click_y)
```

```
169  {
170    XEvent event;
171
172    flame_flush_display(fo);
173
174    while (1)
175      {
176        if (XPending(fo->display) > 0)
177      {
178        XNextEvent(fo->display, &event);
179
180        if (event.type == KeyPress)
181          {
182            /* printf("%c\n", XLookupKeysym(&event.xkey, 0)); */
183            return XLookupKeysym(&event.xkey, 0);
184          }
185        else
186          if (event.type == ButtonPress)
187            {
188          *click_x = event.xkey.x;
189          *click_y = event.xkey.y;
190
191          //Left click == 1, Right click == 3
192          return event.xbutton.button;
193            }
194      }
195        else
196      {
197        ;
198      }
199      }
200  }
201
202
```