

```
1  #include <iostream>
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <memory.h>
5
6  #include <X11/Xlib.h>
7  #include <X11/Xutil.h>
8  #include <X11/XKBlib.h>
9  #include <X11/keysym.h>
10
11 //namespaces
12 using namespace std;
13
14 //global variable declarations
15 bool bFullscreen=false;
16 Display *gpDisplay=NULL;
17 XVisualInfo *gpXVisualInfo=NULL;
18 Colormap gColormap;
19 Window gWindow;
20 int giWindowWidth=800;
21 int giWindowHeight=600;
22
23 //entry-point function
24 int main(void)
25 {
26     //function prototypes
27     void CreateWindow(void);
28     void ToggleFullscreen(void);
29     void uninitialize();
30
31     //variable declarations
32     int winWidth=giWindowWidth;
33     int winHeight=giWindowHeight;
34
35     //code
36     CreateWindow();
37
38     //Message Loop
39     XEvent event;
40     KeySym keysym;
41
42     while(1)
43     {
44         XNextEvent(gpDisplay,&event);
45         switch(event.type)
46         {
47             case MapNotify:
48                 break;
49             case KeyPress:
50                 keysym=XkbKeycodeToKeysym(gpDisplay,event.xkey.keycode,0,0);
51                 switch(keysym)
52                 {
```

```
53         case XK_Escape:
54             uninitialized();
55             exit(0);
56         case XK_F:
57         case XK_f:
58             if(bFullscreen==false)
59             {
60                 ToggleFullscreen();
61                 bFullscreen=true;
62             }
63             else
64             {
65                 ToggleFullscreen();
66                 bFullscreen=false;
67             }
68             break;
69         default:
70             break;
71     }
72     break;
73 case ButtonPress:
74     switch(event.xbutton.button)
75     {
76         case 1:
77             break;
78         case 2:
79             break;
80         case 3:
81             break;
82         default:
83             break;
84     }
85     break;
86 case MotionNotify:
87     break;
88 case ConfigureNotify:
89     winWidth=event.xconfigure.width;
90     winHeight=event.xconfigure.height;
91     break;
92 case Expose:
93     break;
94 case DestroyNotify:
95     break;
96 case 33:
97     uninitialized();
98     exit(0);
99 default:
100     break;
101 }
102 }
103
104 uninitialized();
```

```
105     return(0);
106 }
107
108 void CreateWindow(void)
109 {
110     //function prototypes
111     void uninitialize(void);
112
113     //variable declarations
114     XSetWindowAttributes winAttribs;
115     int defaultScreen;
116     int defaultDepth;
117     int styleMask;
118
119     //code
120     gpDisplay=XOpenDisplay(NULL);
121     if(gpDisplay==NULL)
122     {
123         printf("ERROR : Unable To Open X Display.\nExiting Now...\n");
124         uninitialize();
125         exit(1);
126     }
127
128     defaultScreen=XDefaultScreen(gpDisplay);
129
130     defaultDepth=DefaultDepth(gpDisplay,defaultScreen);
131
132     gpXVisualInfo=(XVisualInfo *)malloc(sizeof(XVisualInfo));
133     if(gpXVisualInfo==NULL)
134     {
135         printf("ERROR : Unable To Allocate Memory For Visual Info.\nExiting
136             Now...\n");
137         uninitialize();
138         exit(1);
139     }
140     XMatchVisualInfo
141         (gpDisplay,defaultScreen,defaultDepth,TrueColor,gpXVisualInfo);
142     if(gpXVisualInfo==NULL)
143     {
144         printf("ERROR : Unable To Get A Visual.\nExiting Now...\n");
145         uninitialize();
146         exit(1);
147     }
148
149     winAttribs.border_pixel=0;
150     winAttribs.background_pixmap=0;
151     winAttribs.colormap=XCreateColormap(gpDisplay,
152         RootWindow(gpDisplay, gpXVisualInfo->screen),
153         gpXVisualInfo->visual,
154         AllocNone);
155     gColormap=winAttribs.colormap;
```

```
155
156     winAttribs.background_pixel=BlackPixel(gpDisplay,defaultScreen);
157
158     winAttribs.event_mask= ExposureMask | VisibilityChangeMask | ButtonPressMask
159         | KeyPressMask | PointerMotionMask |
160         StructureNotifyMask;
161
162     styleMask=CWBorderPixel | CWBackPixel | CWEventMask | CWColormap;
163
164     gWindow=XCreateWindow(gpDisplay,
165         RootWindow(gpDisplay,gpXVisualInfo->screen),
166         0,
167         0,
168         giWindowWidth,
169         giWindowHeight,
170         0,
171         gpXVisualInfo->depth,
172         InputOutput,
173         gpXVisualInfo->visual,
174         styleMask,
175         &winAttribs);
176
177     if(!gWindow)
178     {
179         printf("ERROR : Failed To Create Main Window.\nExiting Now...\n");
180         uninitialize();
181         exit(1);
182     }
183
184     XStoreName(gpDisplay,gWindow,"First XWindow");
185
186     Atom windowManagerDelete=XInternAtom(gpDisplay,"WM_DELETE_WINDOW",True);
187     XSetWMProtocols(gpDisplay,gWindow,&windowManagerDelete,1);
188
189     XMapWindow(gpDisplay,gWindow);
190 }
191
192 void ToggleFullscreen(void)
193 {
194     //variable declarations
195     Atom wm_state;
196     Atom fullscreen;
197     XEvent xev={0};
198
199     //code
200     wm_state=XInternAtom(gpDisplay,"_NET_WM_STATE",False);
201     memset(&xev,0,sizeof(xev));
202
203     xev.type=ClientMessage;
204     xev.xclient.window=gWindow;
205     xev.xclient.message_type=wm_state;
206     xev.xclient.format=32;
207     xev.xclient.data.l[0]=bFullscreen ? 0 : 1;
```

```
206
207     fullscreen=XInternAtom(gpDisplay,"_NET_WM_STATE_FULLSCREEN",False);
208     xev.xclient.data.l[1]=fullscreen;
209
210     XSendEvent(gpDisplay,
211               RootWindow(gpDisplay,gpXVisualInfo->screen),
212               False,
213               StructureNotifyMask,
214               &xev);
215 }
216
217 void uninitialize(void)
218 {
219     if(gWindow)
220     {
221         XDestroyWindow(gpDisplay,gWindow);
222     }
223
224     if(gColormap)
225     {
226         XFreeColormap(gpDisplay,gColormap);
227     }
228
229     if(gpXVisualInfo)
230     {
231         free(gpXVisualInfo);
232         gpXVisualInfo=NULL;
233     }
234
235     if(gpDisplay)
236     {
237         XCloseDisplay(gpDisplay);
238         gpDisplay=NULL;
239     }
240 }
241
242
```