

Complément Xlib – C. Drocourt

1) La fonction XCreateSimpleWindow

```
Window XCreateSimpleWindow (display, parent, x, y, width, height, border_width,  
border_color, background);
```

```
Display *display; /* Le display */  
Window parent; /* La fenetre */  
int x, y; /* La position */  
unsigned int width, height, border_width; /* La taille */  
unsigned long border_color, background; /* Couleur de bord, couleur de fond */
```

2) Affichage des fenêtres

Le fonction `XMapWindow([display],[fenetre])` permet l'affichage de la fenêtre, la fonction réciproque est `XUnmapWindow([display],[fenetre])`.

On pourra lui donner un nom en utilisant la fonction `int XStoreName(Display *display, Window w, char *window_name);`

3) Les différents évènements

Ce sont :

- `ButtonPress` : appui sur le bouton de la souris
- `ButtonRelease` : relachement du bouton souris
- `MotionNotify` : Déplacement de la souris
- `KeyPress` : touche appuyée
- `KeyRelease` : touche relâchée
- `Expose` : Le fenêtre (ou une partie) est exposé a nouveau.

On indique l'interet d'une fenetre pour un évènement particulier en utilisant la fonction :

```
XSelectInput(Display * display, Window w, long event_mask);
```

où `event_mask` est une combinaison de `ButtonPressMask`, `ButtonReleaseMask`, `KeyPressMask`, `KeyReleaseMask`, `ExposureMask`, `PointerMotionMask` ... (la combinaison s'obtient en utilisant le caractère |).

L'attente d'un évènement est réalisé par un appel à la fonction bloquante :

```
int XNextEvent(Display *display, XEvent *event_return);
```

où l'élément `event_return` est une union en C, et dont le premier élément de cette union (le champ `type`) nous donne le type d'évènement généré.

Dans le cas du bouton de souris appuyé (attente de l'évènement `ButtonPress`), on pourra récupérer les coordonnées du pointeur par les variables `objet_event.xbutton.x` et `objet_event.xbutton.y`, pour un déplacement de la souris ce sera `objet_event.xmotion.x` et `objet_event.xmotion.y`.

Pour le clavier, c'est un peu plus compliqué, en effet, il faudra faire un appel à la fonction :

```
int XLookupString(XKeyEvent *event_struct, char *buffer_return,  
int bytes_buffer, KeySym *keysym_return, XComposeStatus *status_in_out);
```

On passe le pointeur sur l'évènement et on récupère le keysym de la touche enfoncée, ainsi qu'une chaîne de caractère contenant cette touche (on positionnera en général le dernier paramètre à 0). Le keysym est un code spécifique à X Window, qui vaudra par exemple :

XK_space : espace,
XK_a : la lettre a,
XK_0 : la lettre 0,
XK_Down : la flèche bas,
XK_Return : la touche entrée,
...

4) Le contexte graphique

On peut créer un GC (contexte graphique) en utilisant la fonction :

```
GC XCreateGC(Display *display, Drawable d, unsigned long valuemask,  
             XGCValues *values);
```

(Des exemples se trouvent dans le cours précédent).

5) Afficher dans la fenêtre

Vous pouvez utiliser les fonctions suivantes :

```
int XClearWindow(Display *display, Window w);  
int XDrawString(Display *display, Drawable d, GC gc, int x, int y, char  
*string, int length);  
int XDrawLine(Display *display, Drawable d, GC gc, int x1, int y1, int x2,  
int y2);  
int XDrawPoint(Display *display, Drawable d, GC gc, int x, int y);  
int XDrawRectangle(Display *display, Drawable d, GC gc, int x, int y,  
unsigned int width, unsigned int height);  
int XDrawArc(Display *display, Drawable d, GC gc, int x, int y, unsigned  
int width, unsigned int height, int angle1, int angle2);
```