

Exercice 1 : un click

On souhaite créer un client X avec une fenêtre principale contenant deux filles (une avec un fond rouge et l'autre avec un fond noir).

A chaque instant, une seule fille est visible.

En cliquant sur la fille visible, elle disparaît en faisant apparaître l'autre fille.

On considère qu'un clic est composé d'un évènement ButtonPress et d'un ButtonRelease espacés au maximum de 3 secondes. Au delà de 3 secondes, ils seront simplement ignorés.

1. Compléter le programme en sélectionnant le ou les évènements utiles pour ces clics.
2. Mettre en place le ou les actions énoncées ci-dessus en ignorant le temps écoulé.
3. En considérant les 3 secondes de délai.

Exercice 2 : à la poursuite de la souris

On souhaite créer un client X avec une fenêtre principale contenant une fenêtre fille de fond noir ayant une largeur et une hauteur correspondant à 1/10^{ème} de celles de sa mère.

Quelque soit le changement de taille de la fenêtre principale, la fenêtre fille doit garder toujours sa taille à 1/10^{ème} de celle de sa mère.

On souhaite pouvoir déplacer la fenêtre fille à l'intérieur de sa mère :

- On enfonce un bouton dans la fenêtre fille.
- Sans relâcher le bouton, on déplace la souris et la fenêtre fille suit le déplacement.
- Pour la déposer, on relâche le bouton.
- Lors du déplacement, tous les autres évènements concernant les autres boutons n'a aucun effet. Pas d'effet parasite.
- Quelque soit le déplacement, la fenêtre fille doit rester entière autant que possible dans la fenêtre de sa mère.

- On ne pourra la déplacer à nouveau qu'après avoir lâché tous les boutons.
1. Compléter le programme en sélectionnant le ou les événements utiles pour l'opération indiquée ci-dessus.
 2. Mettre en place le ou les actions énoncées ci-dessus.

Exercice 3 : la roulette

Dans X Window System, la souris est prévue accompagner de 5 boutons. On a habituellement une souris avec 3 boutons et une roulette. Cette dernière remplace les boutons 4 et 5 de la manière suivante :

- la roulette fonctionne avec des crans,
- elle a une certaine inertie : en la roulant avec une certaine force et sans l'arrêter, elle ralentit avant de s'arrêter toute seule.
- en la roulant de l'avant vers l'arrière (resp. de l'arrière vers l'avant), chaque cran émet un `ButtonPress` suivi juste après par un `ButtonRelease` correspondant au bouton 5 (resp. bouton 4).

Puisque ces deux événements vont dans ce cas par pair pour les boutons 4 et 5, on propose de traiter uniquement le `ButtonRelease`.

On souhaite créer une application affichant une fenêtre principale de fond noir avec les comportements suivants :

- En cliquant sur un des boutons 1, 2 ou 3, la fenêtre devient toute noire.
- En tournant la roulette de l'avant vers l'arrière (bouton 5), le fond de la fenêtre rougit dont l'intensité augmente avec le nombre de `ButtonRelease` reçu.
- En tournant la roulette de l'arrière vers l'avant (bouton 4), le fond de la fenêtre bleuit dont l'intensité augmente avec le nombre de `ButtonRelease` reçu.

1. Compléter le programme en sélectionnant le ou les événements utiles pour l'opération indiquée ci-dessus.
2. Y créer un contexte graphique adapté à la fenêtre principale et que l'on peut modifier la couleur du foreground en bleu ou en rouge avec une certaine intensité.

On propose

- d'incrémenter d'abord le compteur de 1 à chaque `ButtonRelease` reçu.
- de prendre arbitrairement la valeur de l'intensité de la couleur comme étant `compteur * 20` (valeur à ajuster).

3. Compléter la procédure `PourButtonRelease5()` pour pouvoir afficher la couleur rouge comme indiquer ci-dessus.

On propose enfin de remettre le compteur à zéro et le fond de principale à noir à la fin de chaque roulement ou lors d'un arrêt intentionnel de la roulette.

4. Compléter le programme pour cela.

Annexe

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

Display *XOpenDisplay (char *display_name) ;

XCloseDisplay (Display *display) ;

Window DefaultRootWindow (Display *display);

int DefaultScreen (Display *display);

Window XCreateWindow (Display *display,
    Window parent, int x, int y, unsigned int largeur,
    unsigned int hauteur, unsigned int largeur_bord,
    int profondeur, unsigned int classe, Visual *type_visuel,
    unsigned long masque, XSetWindowAttributes *attributs) ;

Window XCreateSimpleWindow (Display *display,
    Window parent, int x, int y, unsigned int largeur,
    unsigned int hauteur, unsigned int largeur_bord,
    unsigned long bordure, unsigned long arrière-plan) ;

XSelectInput (Display *display,
    Window f, long masque_événements)

int XStoreName (Display *display,
    Window w, char *window_name);

XMapWindow (Display *display, Window f) ;

XMapSubwindows (Display *display, Window f) ;

XUnmapWindow (Display *display, Window f) ;

XConfigureWindow (Display *display,
    Window f, unsigned int masque, WindowChanges *valeurs) ;

XMoveWindow (Display *display, Window f, int x, int y)

XResizeWindow (Display *display,
    Window f, unsigned int largeur, unsigned int hauteur) ;

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

int XSetForeground(Display *display, GC gc,
    unsigned long foreground);

int XfillRectangle (Display *display, Drawable d, GC gc,
    int x, int y, unsigned int width, unsigned int height);
```