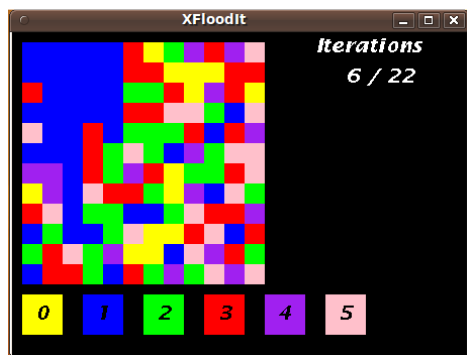


ZZ1 C TP n°5

Xlib : Interface graphique du jeu FloodIt

Vous aller ajouter une couche graphique au jeu dont vous avez développé le moteur texte au TP précédent. Veuillez à bien sauvegarder le(s) fichier(s) du moteur.



Vous trouverez sur <http://fc.isima.fr/~loic> un exemple de fichier xlib. Ce programme que l'on vous propose se découpe en phases classiques d'un programme xlib

- Déclarations et Initialisations
- Connexion graphique et initialisations graphiques spécifiques
- Boucle principale pour la gestion des événements
- Libération des ressources.

Ce programme n'est pas exempt de défauts au niveau de l'implémentation : toutes les variables sont globales par commodité et la libération des ressources n'est pas exhaustive.

Regardez bien ce que fait ce programme : il vous montre comment afficher des formes simples, comment récupérer les événements claviers, changer de police de caractères et manipuler la couleur

Gestion de la couleur

Gérer les couleurs en Xlib est un peu pénible à cause de l'âge de la bibliothèque et de la foule de matériels différents à prendre en compte. Les capacités du display sont connues grâce à la notion de visuels (Visual) : un visuel est proposé par défaut (c'est celui de la fenêtre racine) mais il est possible d'en changer. Pour avoir la liste de visuels disponibles et celui qui est proposé par défaut, soit vous tapez la commande shell **dpypinfo**, soit vous appelez la fonction Xlib idoine.

Les visuels les plus intéressants pour nous sont le DirectColor et le TrueColor avec des couleurs stockées sur 24 bits (8bits pour chaque composante rouge, vert et bleue. Attention, cela peut être moins). La palette de couleurs est en lecture-écriture pour le premier visuel et en lecture seule pour le second, le fait qu'elle soit en lecture seule est un avantage (cela permet entre autres de partager la

palette entre les différentes applications graphiques tournant sur le serveur X). Nous supposons que le visuel renvoyé est de type TrueColor (En toute rigueur, il faudrait vérifier cette supposition et utiliser le bon visuel dans le cas échéant).

Nous allons demander une couleur à la palette grâce à son nom (en Anglais, bien sur). La liste des couleurs se trouve dans un fichier rgb.txt dans un sous-répertoire des inclusions. La structure retournée est de type XColor et on donne le champ pixel à la fonction XSetForeground() pour changer de couleur.

Implémentation

Après ces brèves explications, passons aux choses sérieuses.

- Copier le fichier d'exemple et changer de nom. Vous allez travailler à ajouter / modifier des éléments dans ce fichier pour ne pas avoir à tout retaper.
- Je vous conseille de mettre les variables spécifiques au jeu dans le lot des variables globales (grille et itérations). Vous devez également faire un tableau de couleurs contenant le champ pixel des XColor et de charger une police de caractères avec une certaine taille.
- La méthode afficheGraphisme() appellera afficheGrille(), on pourra aussi afficher l'itération courante et les raccourcis clavier pour les couleurs, comme sur la copie écran proposée. L'affichage de la grille se fera par l'affichage de SIZE*SIZE carrés de couleur.
- Dans la méthode saisie(), on gèrera les entrées clavier pour le changement de couleur et les autres choix claviers (sortie, ...). La variable de fin de boucle est déjà définie, il n'y a plus qu'à s'en servir. Pour transformer un nombre en une chaîne de caractères ou inversement, vous pouvez utiliser les fonctions sprintf() et sscanf().
- Bien vérifier que les ressources sont bien libérées. valgrind peut envoyer des résultats intéressants

Voilà un site intéressant sur la Xlib : <http://tronche.com/gui/x/xlib/>

Pour aller plus loin :

- On peut imaginer choisir la couleur d'autres façons : clic sur une cellule de la grille ou sur les couleurs du bas de l'écran ☺. C'est facile, on connaît leur position. Dans le cas d'un événement de type ButtonPress, le champ xbutton de la structure XEvent a un sens et on accède à la coordonnée x en faisant ev.xbutton.x.
- On peut essayer de se débarrasser des variables globales
- Utiliser la compilation séparée pour différencier les éléments : moteur de jeu, interface texte, interface graphique ...