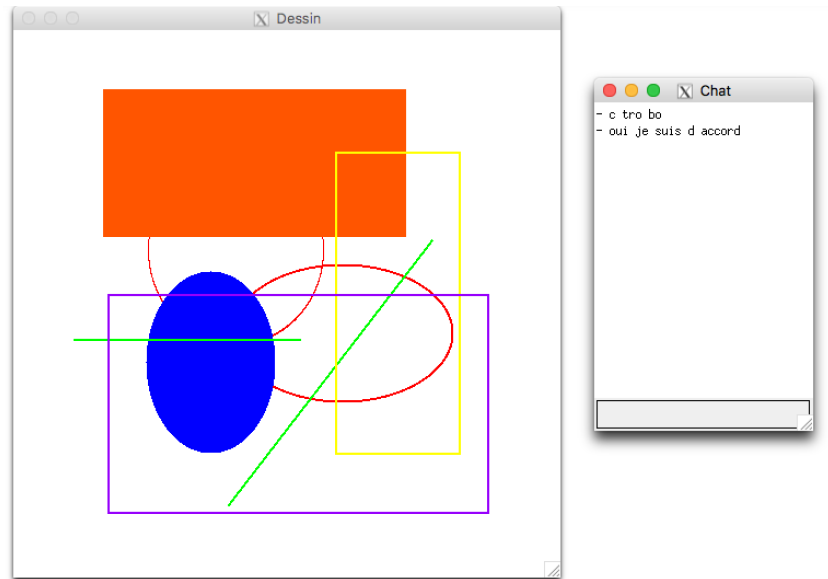


# 2I007 - Projet n° 1 : Dessin collaboratif

## VERSION 0.1 (subira peut être des modifications)

Ce projet consiste à réaliser un programme X Window System permettant de dessiner sur le réseau : tous les clients connectés partageront une même planche à dessin, qui sera mise à jour dès lors que quelqu'un y dessinera. On pourra, dans un menu, choisir une forme, une couleur, et une épaisseur de trait pour les dessins. Un chat sera également présent pour permettre aux utilisateurs de discuter de la portée artistique de leurs œuvres communes.

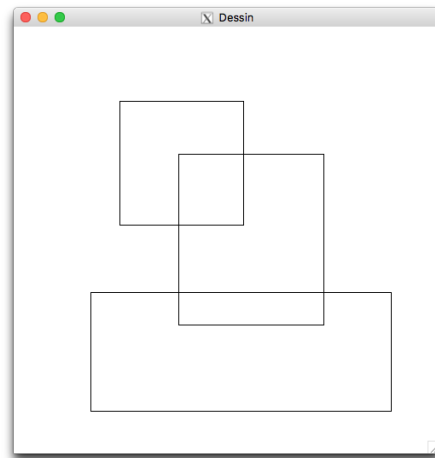


Pour réaliser cette application, vous complétez le fichier “canevas” qui vous sera remis de manière électronique.

### 1 Interface

En premier lieu, on va se contenter de créer les éléments d'interface et de gérer les événements pour dessiner dans la planche à dessin. On gèrera la communication entre les différents clients par la suite.

## 1.1 La planche à dessin



La planche à dessin est une simple fenêtre dans laquelle on va dessiner. Ces dessins sont des formes prédéfinies (rectangle vide, rectangle plein, ellipse vide, ellipse pleine, ligne, ...) et seront créés en utilisant la souris (voir explications ci-dessous).

Lorsqu'on appuie sur le bouton gauche de la souris dans la planche à dessin, on commence à dessiner : le point  $(x1,y1)$  est enregistré comme point de départ de la forme, et, en se déplaçant tout en gardant le bouton de la souris appuyé, on va allonger la forme jusqu'à son point d'arrivée  $(x2,y2)$  qui sera enregistré lorsqu'on relâchera le bouton. Par exemple, le point  $(x1,y1)$  pour un rectangle est un coin de départ et le point  $(x2,y2)$  est son coin opposé.

Pendant la construction avec le déplacement de la souris, on souhaite pouvoir observer un aperçu "élastique" de la forme que l'on est en train de dessiner (on se servira d'un GC en mode XOR pour dessiner les traits élastiques).

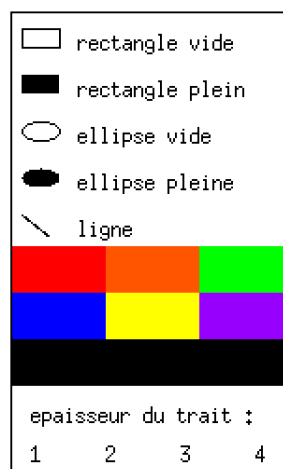
Une fois le bouton relâché, la forme est dessinée de manière permanente sur la fenêtre.

### À faire :

Vous sensibiliserez la fenêtre principale aux événements souris attendus, puis vous traiterez ces derniers dans les fonctions qui leur sont associées (et qui seront appelées dans la boucle d'événements).

Vous complétez en particulier la fonction Draw qui permettra de dessiner une forme avec un GC donné, ainsi que sa couleur, son épaisseur, et sa position passées en paramètre.

## 1.2 Le menu



Le menu de l'application est une fenêtre, d'abord invisible, ne possédant pas de décorations issues du Window Manager.

Lorsque l'utilisateur appuiera sur le bouton droit de la souris, le menu de l'application s'affichera juste à côté de son curseur. En déplaçant le curseur (tout en restant appuyé sur le bouton) vers les divers éléments du menu, l'utilisateur pourra choisir une forme de dessin, une couleur, ou une épaisseur de trait.

Lorsque le bouton droit sera relâché, alors le menu disparaîtra, et l'attribut qui se trouvait sous le curseur sera sélectionné (par exemple, si on relâche la souris pendant que le curseur est sur le carré rouge, alors la forme que l'on s'apprête à dessiner sera de couleur rouge).

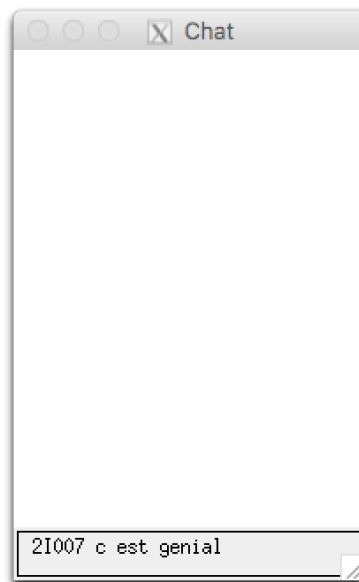
#### À faire :

Vous tâcherez de faire en sorte que le menu ressemble (à peu près) à la capture d'écran affichée ci-dessus.

Vous sensibiliserez le menu (et ses fenêtre filles) aux événements correspondant à ce comportement, et traiterez ces événements dans des fonctions associées.

Selon l'élément sur lequel on a arrêté la souris, vous enregistrerez la nouvelle forme, couleur, ou épaisseur de trait dans leurs variables respectives.

### 1.3 Fenêtre de chat



Une fenêtre de chat sera également affichée séparément de la fenêtre de dessin. Cette fenêtre sera sensible aux événements du clavier, et mettra à jour une chaîne de caractères à chaque fois que l'utilisateur tapera sur une touche du clavier (cela aura pour effet d'ajouter la lettre correspondante à la fin de la chaîne). Cette chaîne sera affichée dans une petite zone de texte située en bas de la fenêtre.

#### À faire :

Vous sensibiliserez la fenêtre de chat aux événements correspondant à ce comportement, et traiterez ces événements dans des fonctions associées. Vous tâcherez en particulier de gérer convenablement l'appui sur la touche espace du clavier, ainsi que sur la touche de retour arrière (BackSpace) pour effacer la dernière lettre tapée.

Une fois le message tapé, il sera envoyé sur le réseau par un appui sur la touche Entrée du clavier.

## 2 Communications

Une fois, et une fois seulement, que les éléments d'interface auront l'apparence et le comportement attendus, on peut s'attaquer aux communications entre les clients.

### 2.1 Envoi de formes

Il y a 5 informations à transmettre qui sont nécessaires pour reproduire un dessin dans un autre client :

- La forme
- La couleur
- L'épaisseur du trait
- Le point de départ de la forme (x1,y1)
- Le point d'arrivée de la forme (x2,y2)

Ces informations vont être transmises par l'intermédiaire de propriétés : à chaque fois que l'on aura fini de dessiner (i.e. que le bouton gauche de la souris sera relâché), on va modifier ces propriétés. Cette modification alertera tous les clients à l'écoute du changement de propriété (qui pourront ainsi mettre à jour leur application avec la ou les nouvelles valeurs sélectionnées).

#### À faire :

Vous adapterez le code pour que chaque propriété soit modifiée dès lors que le bouton gauche de la souris est relâché.

Une fois qu'un client aura été prévenu de la modification de l'ensemble des propriétés permettant de reproduire une forme, alors cette forme sera affichée sur sa planche à dessin.

Vous sensibiliserez les fenêtres concernées aux changements sur ces propriétés.

Vous traiterez de manière adéquate l'arrivée d'une notification vous informant d'un changement de propriétés, et appellerez la fonction Draw dès lors que toutes les informations nécessaires à l'affichage de la forme auront été récupérées.

Il est à noter que le client à l'origine du changement de propriété va lui aussi être prévenu de cette modification, et donc afficher aussi la forme. Ce n'est pas forcément très gênant mais, pour éviter de dessiner par deux fois la dernière forme, on va simplement retirer l'appel à la fonction Draw lors du relâchement de la souris : c'est la réaction à notre propre modification des valeurs des propriétés qui entraînera l'affichage de la forme qu'on a dessinée.

### 2.2 Chat

Les messages dans le chat seront également transmis par l'intermédiaire d'une propriété : lorsqu'on appuiera sur la touche Entrée du clavier après avoir écrit dans la zone de texte du chat, une propriété XA\_CHAT sera modifiée, et tous les clients seront prévenus de ce changement. Leur réaction sera d'afficher dans leur propre fenêtre de chat chaque nouveau message reçu, les uns en dessous des autres.

#### À faire :

Vous sensibiliserez les fenêtres concernées à ce comportement, et modifierez le code pour traiter le nouvel événement. Pour soucis de simplicité, une fois qu'il n'y aura plus de place dans la fenêtre de chat, celle-ci sera simplement effacée.

## 3 Bonus

Vous avez le champ libre pour ajouter des fonctionnalités supplémentaires à vos applications. Pour vous inspirer, nous vous proposons quelques exemples de fonctionnalités intéressantes :

### **3.1 Réglettes RGB**

Plutôt que choisir une couleur fixe dans le menu de l'application, il pourrait être intéressant d'afficher dans la fenêtre de dessin trois réglettes rouge, vert et bleu permettant de choisir précisément la couleur que l'on souhaite utiliser pour dessiner des formes.

### **3.2 Dessin à main levée**

Au lieu de se limiter à des formes prédéfinies, il serait amusant de pouvoir aussi dessiner directement dans la planche à dessin comme avec un crayon.

### **3.3 Pseudos et couleurs dans le chat**

Tel quel, le chat est très minimaliste. Il pourrait être utile d'ajouter la possibilité pour les clients de choisir un pseudo (dans une zone de texte), et même de pouvoir décider de la couleur de leurs messages.

## **4 Rendu du projet**

Ce projet pourra s'effectuer en monôme ou binôme, et devra impérativement être rendu avant le 7 mars 23h59 pour le groupe 1 et le 9 mars 23h59 pour les groupes 2 et 3, sous la forme d'un fichier `NOM1_NOM2.tar.gz` envoyé par e-mail (avec pour objet "[2I007 - Projet] NOM1 NOM2") à votre chargé de TD et contenant l'intégralité des fichiers `.c` nécessaires pour compiler votre application.

En cas d'ajout de fonctionnalités, veuillez les décrire rapidement dans le mail.