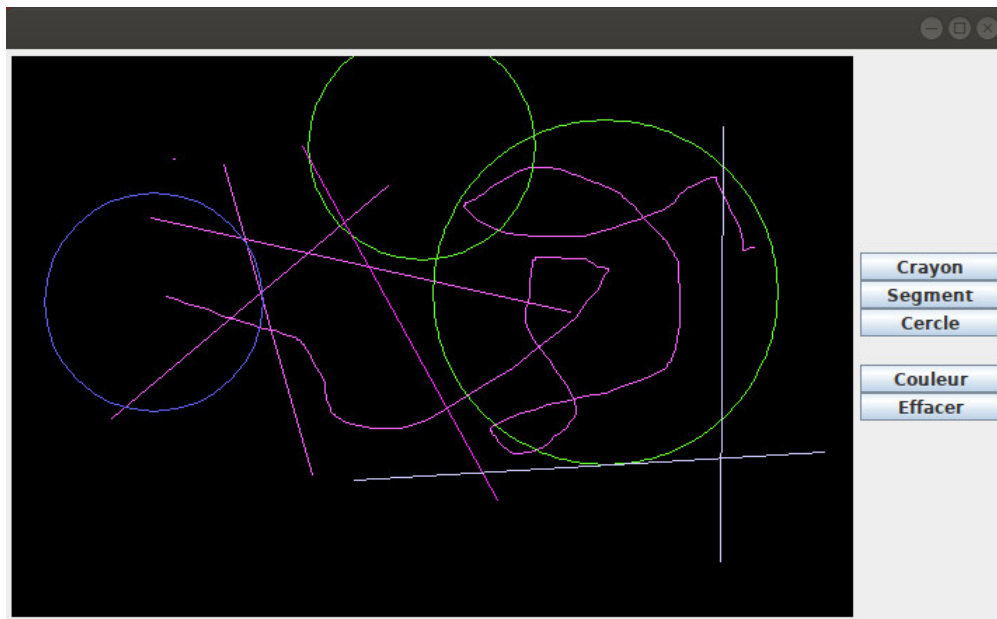


## Interfaces Graphiques

### TP 2 – Mini-Gimp



Ce TP peut être réalisé en deux semaines. Le but sera de réaliser une application permettant de dessiner à la souris dans une fenêtre, sur le modèle de la capture d'écran ci-dessus.

À gauche de l'écran se trouve l'*ardoise*. À droite se trouve la *boîte à outils*. Les boutons "Crayon", "Segment" et "Cercle" permettent de sélectionner la manière dont seront interprétés les événements-souris reçus par l'ardoise.

- L'outil "Crayon" permet de dessiner en tracé libre à l'écran en maintenant appuyé le bouton de la souris.
- L'outil "Segment" permet de dessiner des segments en appuyant sur le bouton de la souris pour choisir sa première extrémité, en maintenant le bouton appuyé et en déplaçant la souris, puis en relâchant le bouton. La seconde extrémité est le point de relâchement.
- L'outil "Cercle" permet de dessiner des cercles en appuyant sur le bouton de la souris pour choisir son centre, en maintenant le bouton appuyé et en déplaçant la souris, puis en relâchant le bouton. Le point de relâchement est interprété comme un point du cercle.

Toujours dans la boîte à outils, le bouton "Couleur" fait monter à l'écran un sélecteur de couleur en mode "dialogue", permettant de choisir la couleur du tracé. Le bouton "Effacer" permet, bien sûr, d'effacer l'image interne de l'ardoise.

## Conseils de réalisation

Les éléments suivants ne sont qu'indicatifs et ne s'appliquent qu'à la correction de ce mini-projet telle qu'elle a été écrite.

1. L'assemblage de l'application se fait dans le constructeur de son panneau principal, instance d'une classe héritière de `JPanel`. Le `main` se contente de créer une instance de cette classe et de la placer dans une `JFrame`.
2. L'*ardoise* est définie comme une extension de `JPanel`. Le dessin ne se fait pas directement dans l'ardoise mais dans une `BufferedImage` de type RGB encapsulée dans celle-ci. Le procédé permet de préserver les pixels dessinés si la fenêtre est recouverte puis redécouverte.
3. La méthode `protected void paintComponent(Graphics g)` de l'ardoise a été redéfinie comme affichant l'image interne de l'ardoise à chaque demande de rafraîchissement – en particulier à chaque invocation de `repaint()` sur l'ardoise, qui invoquera en réponse sa méthode `paintComponent` en lui fournissant automatiquement un `Graphics` servant de point d'accès à ses pixels.
4. L'écoute des événements-souris dans l'ardoise est sous la responsabilité d'instances de trois classes de contrôleurs, héritières d'une classe abstraite de contrôleurs, elle-même héritière de `MouseAdapter`. La classe abstraite encapsule un champ de type *ardoise*, un autre de type `String` – le nom d'un contrôleur – et contient deux méthodes `void activer()` et `void desactiver()`. La première permet d'ajouter un contrôleur à la liste des `MouseListener` et `MouseMotionListener` de l'ardoise encapsulée, la seconde de le supprimer de ces deux listes.
5. La boîte à outils est définie elle-aussi comme une extension de `JPanel`. Elle contient un champ de type contrôleur abstrait, le *contrôleur courant*. Les boutons sont construits par la boîte elle-même et ajoutés à celle-ci via deux méthodes :
  - (a) La première prend en argument un `String` et un `ActionListener` et crée un bouton étiqueté par le `String` et écouté par cet `ActionListener`. C'est cette méthode qui est utilisée pour ajouter les boutons "Couleur" et "Effacer".
  - (b) La seconde prend en argument un contrôleur abstrait et un booléen. Elle crée un bouton étiqueté par le nom de ce contrôleur, et un `ActionListener` dont l'effet est de : désactiver le contrôleur courant ; activer le contrôleur argument ; remplacer le contrôleur courant par le contrôleur argument. Le booléen indique si le champ doit être initialisé par le contrôleur argument – il est utilisé pour activer le contrôleur "Crayon" au lancement de l'application.
6. Le bouton "Couleur" fait monter à l'écran un `JColorChooser` en mode "dialogue".
7. Dans les modes "Segment" ou "Cercle", le segment ou le cercle candidat à être tracé est affiché et effacé en continu pendant le "drag" de souris, *c.f.* le "XOR mode" de `Graphics`.

## Si vous ne savez plus quoi faire

Ajoutez un menu permettant de quitter l'application, de charger une image dans l'ardoise, et de sauvegarder l'image dessinée.