

# Enseignement de spécialité Informatique et sciences du numérique Formation des IA-IPR et chargés de mission Atelier de programmation 3

David Pichardie, Luc Bougé

Mardi 15 mars

## Activité I. Réalisation d'un logiciel de manipulation d'images

Utiliser la *proglet* Javascool *Dessiner sur une image*. Ce projet vous propose de compléter ce logiciel de dessin bien connu avec différentes fonctionnalités basiques comme le tracé de courbes, de lignes droites, de rectangles ou encore le remplissage de forme.

Prendre connaissance des fonctionnalités attendues pour ce logiciel en utilisant la *proglet* en mode *Démo* (option à cocher en bas à droite de l'onglet *Paintbrush*). L'option *Hexa code* vous montre que cette image est en fait constituée de 32 pixels de large sur 32 pixels de haut (soit 1024 pixels), chaque pixel contenant une couleur encodée sous la forme d'un nombre entre 0 et 15. Dans ce mode, une couleur est représentée par son numéro au format hexadécimal :

1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

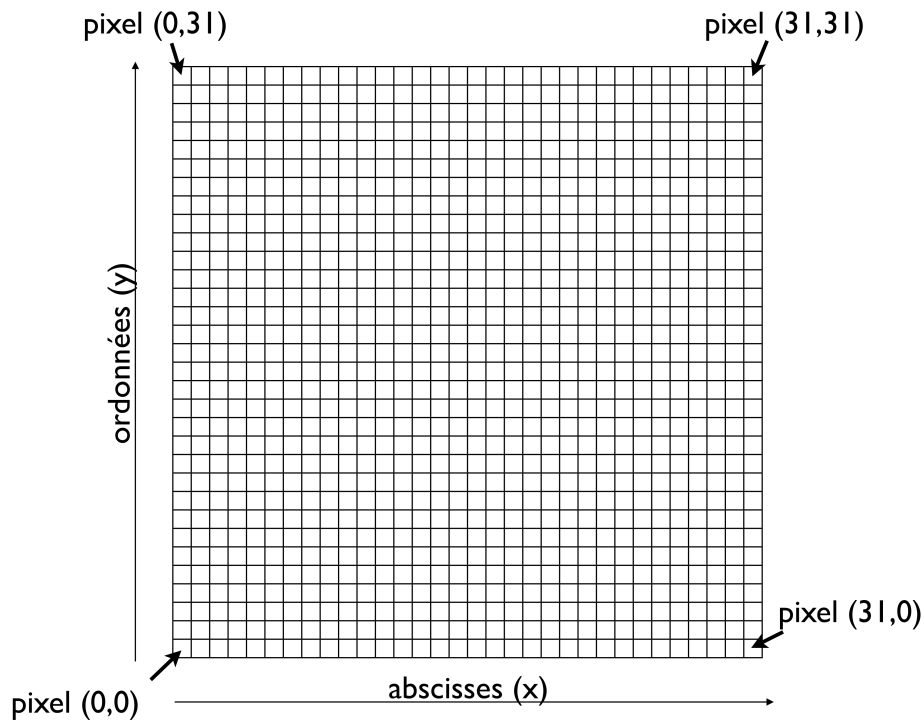
Observer la mise à jour des codes numériques de chaque pixel lors du remplissage d'une forme par exemple.

Passer ensuite en mode *Proglet*, les fonctionnalités précédentes ne fonctionnent plus ! Vous devez vous-même programmer ces fonctionnalités. Il faut pour cela recopier le squelette de programme fourni (voir l'énoncé de cette séance dans Javascool) et compléter chaque fonction.

Afin de modifier les pixels de l'image, vous disposez de l'objet `image` sur lequel deux actions sont possibles :

- `getPixel(x,y)` renvoie le code couleur du pixel de coordonnées  $(x,y)$ .
- `setPixel(x,y,couleur)` modifie le pixel de coordonnées  $(x,y)$  avec la couleur de numéro couleur.

Chaque abscisse et chaque coordonnée doit varier dans l'intervalle  $[0, 31]$ . Une erreur se déclenchera si vous utilisez ces fonctions hors de ces bornes.



**Exercice I.1.** Compléter la fonction `affichePoint` qui est utilisée par le logiciel en mode *Tracé* à chaque fois qu'il est nécessaire de colorier un pixel.

**Exercice I.2.** Compléter la fonction `supprimePoint` qui est utilisée par le logiciel en mode *Gomme* pour effacer une zone carrée de 3 pixels sur 3 pixels. Effacer un pixel revient à le colorier avec la couleur de fond (ici le blanc).

**Exercice I.3.** Compléter la fonction `afficheRectangle` qui est utilisée par le logiciel en mode *Rectangle*. L'utilisateur désigne un premier point, déplace la souris vers un deuxième point puis relâche le bouton de la souris. Ces deux points constituent les extrémités de la diagonale du rectangle à tracer. Le logiciel fait alors appel à la fonction `afficheRectangle` en donnant en argument les coordonnées du premier point, les coordonnées du deuxième point ainsi que la couleur demandée pour tracer les contours du rectangle.

**Exercice I.4.** Compléter la fonction `remplir` qui est utilisée par le logiciel en mode *Remplir*. Les coordonnées données en paramètres sont celles du pixel sélectionné par l'utilisateur avec son curseur *pot de peinture*.

**Remarque.** *Attention, cette solution demande un peu de réflexion pour éviter de faire boucler la procédure de remplissage mais il existe une solution relativement simple et concise !*

**Exercice I.5.** Compléter la fonction `afficheLigne` qui est utilisée par le logiciel en mode *Ligne*. L'utilisateur désigne un premier point, déplace la souris vers un deuxième point puis relâche le bouton de la souris. Ces deux points constituent les extrémités de la ligne à tracer. Le logiciel fait alors appel à la fonction `afficheLigne` en donnant en argument les

coordonnées du premier point, les coordonnées du deuxième point ainsi que la couleur demandée pour tracer la ligne.

**Remarque.** *Attention, cette fois les lignes à tracer ne sont pas forcément horizontales ou verticales. Comme le montre la figure ci-dessous, il faut savoir faire des approximations pour allumer les pixels les plus proches du segment idéal.*

