

Fondements de l'informatique graphique

Master ISICG

TP2 : Remplissage de polygones

8 octobre 2014

Au cours de ce TP, nous allons nous attaquer au principe du remplissage de polygone. Vous pouvez effacer votre grille à l'aide de la fonction `effacer()`.

1 Remplissage de triangle

- A l'aide de trois points sélectionnés à la souris, dessinez un triangle en utilisant l'algorithme de Bresenham.
- utilisez l'algorithme de remplissage de triangle quelconque pour remplir ce triangle.

2 Généralisation aux polygones convexes

- Sélectionnez cette fois-ci plusieurs points sur la grille, et, grâce à l'appui d'une touche du clavier, tracez le polygone qui relie ces points.
- Vérifiez la convexité du polygone.
- Utilisez une extension de l'algorithme de remplissage de triangle pour remplir votre polygone par un algorithme de balayage.

3 Remplissage de formes

Le principe va être ici de remplir une forme quelconque dont les bords auront été dessinés à la souris, par une méthode de type *seedfill*.

- Nous commencerons par un premier algorithme récursif simple, appelé une première fois avec les coordonnées d'un *pixel* appartenant à la forme et sélectionné par l'utilisateur :
Si le pixel est éteint, l'allumer (en modifiant sa valeur dans le tableau et en le traçant) et appeler récursivement la fonction sur ses quatre voisins (haut, bas, gauche et droite).
- Ce premier algorithme peut parfois être pénalisant sur la mémoire, en raison du coût important de la pile des appels récursifs. Essayez maintenant un deuxième algorithme, basé sur une file de *pixel* à traiter, toujours initialisée avec un pixel appartenant au polygone : Si le *pixel* à traiter

est éteint, l'allumer, puis allumer tous ses voisins de gauche et de droite, jusqu'à rencontrer un bord du polygone, créant ainsi une ligne de *pixels* allumés contenue dans le polygone. Ajouter ensuite, pour chacun des pixels de cette ligne, chacun des voisins du haut et du bas à la fin de la file d'éléments à traiter. Il suffit ensuite de réitérer l'opération tant que la file n'est pas vide.

- Votre algorithme doit également fonctionner pour des formes comportant des *trous* à l'intérieur.