Département d'Informatique
Master mention informatique spécialité I3D
Première année
Année 2018-2019

Fondements et Algorithmes de l'Imagerie Numérique

Projet : Saisie et remplissage d'un polygone 2D

L'objectif de ce projet est d'écrire un programme en C ou C++ vous permettant de dessiner un polygone plan et de le remplir. Vous pouvez partir du programme fourni en TP appelé Kit d'expérimentations 2D discret et qu'on peut également trouver sur Moodle. Je vous propose de procéder par étapes. Pour l'évaluation, si vous avez fini le projet, il ne sera pas nécessaire de me montrer toutes les étapes, mais seulement le programme final.

1 Afficher une ligne brisée (2p)

Écrire un programme qui, à partir d'un ensemble de coordonnées de points dans le plan (spécifiées directement dans le code source), affiche une ligne brisée qui relie ces points par des droites de Bresenham dans l'ordre de leur spécification.

2 Dessin à la souris (2p)

Écrire un programme où chaque clique de la souris ajoute un nouveau sommet en ce point et, le cas échéant, le relie au sommet précédent. Lorsqu'on presse la touche 'c' (comme dans *close*) la ligne brisée ouverte devient un polygone fermé. Lorsqu'on presse de nouveau la touche 'c', le polygone fermé redevient une ligne brisée ouverte. Dans cette version, il n'est pas nécessaire qu'on puisse insérer des sommets ou en supprimer.

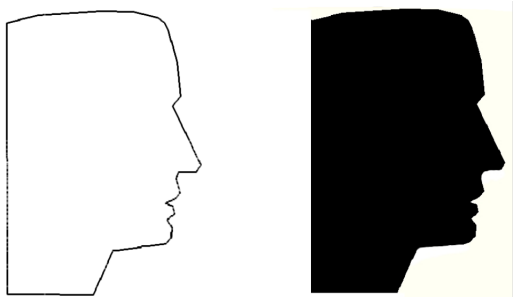

FIGURE 1 – À gauche, un polygone fermé et, à droite, le même polygone rempli par *scan-line.*

3 Remplissage scan-line (6p)

Écrire une nouvelle version du programme précédent qui grâce à la touche 'f' (comme dans *fill*) permet de remplir le polygone fermé avec la méthode *scan-line*. Est-ce que cet algorithme fonctionne avec des polygones non-simples ?

4 Insérer et supprimer des sommets (3p)

Trouver une structure de données adéquate permettant de représenter les lignes brisées et les polygones fermés et qui permette également d'insérer des sommets et d'en supprimer. Quand je dis structure de données adéquate, j'entends que le fait d'insérer ou d'enlever un élément ne doit pas nécessiter de parcourir le tableau pour décaler tous les éléments pour faire de la place.

5 Sélection par clavier (2p)

Écrire une nouvelle version du programme précédent qui admette trois modes :

insert (touche i) Dans le mode *append* le programme fonctionne comme auparavant : chaque clique de la souris avec le bouton gauche crée un nouveau sommet et le relie au dernier point crée.

vertex (touche v) Dans le mode *vertex*, un sommet et un seul est supposé sélectionné. Le sommet sélectionné est visualisé par un signe distinctif (par exemple un petit carré autour du sommet).
- La touche page suivante (respectivement page précédente) permet de sélectionner le sommet suivant (respectivement précédant) le sommet sélectionné.
- Les touches haut, bas, droite, gauche permettent de déplacer le sommet sélectionné dans la direction correspondante.
- La touche suppr permet d'éliminer le sommet sélectionné.

edge (touche e) Dans le mode *edge*, une arête et une seule est supposée sélectionnée. Cette arête est visualisée par un signe distinctif. On peut imaginer qu'elle soit dessinée d'une couleur différente des autres arêtes.
- La touche page suivante (respectivement page précédente) permet de sélectionner l'arête suivant (respectivement précédant) l'arête sélectionnée.
- un clique avec le bouton du milieu permet de couper l'arête sélectionnée en deux en insérant un nouveau sommet entre ses extrémités.

6 Sélection par souris (2p)

Sommet le plus proche Écrire une fonction *closestVertex* qui, étant donnée une position (x, y) et une ligne brisée ou un polygone fermé, retourne l'indice du sommet le plus proche de (x, y).

Arête la plus proche Écrire une fonction *closestEdge* qui, étant donnée une position (x, y) et une ligne brisée ou un polygone fermé, retourne l'indice de l'arête la plus proche de (x, y).

Écrire une nouvelle version du programme précédent qui réalise les mêmes actions sauf qu'on n'utilise plus les touches page précédente et page suivante pour changer les sélections.
- en mode vertex le fait de cliquer avec le bouton gauche revient à sélectionner le sommet le plus proche ;
- en mode edge le fait de cliquer avec le bouton gauche revient à sélectionner l'arête la plus proche. Pour simplifier on admettra que l'arête la plus proche est adjacente au sommet le plus proche.

7 Évaluation

Vous réaliserez ce travail individuellement. Rien ne vous empêche de travailler en groupe, mais au moment de l'évaluation, je m'attends à ce que vous connaissiez le code comme si vous l'aviez écrit seul. La semaine du 5 novembre, lors du TP, je demanderai à chacun d'entre vous de prendre 2 minutes pour me montrer le fonctionnement de son projet en l'état (tester le remplissage avec soin). Ensuite, avant le dimanche 18 novembre à 23h55, chacun complétera son dépôt git
- le code source du projet
- une image (capture d'écran par exemple) me montrant le résultat auquel vous êtes arrivé (peut être incluse dans le README)
- un document README.md décrivant à la lettre les manipulations que vous avez effectuées pour obtenir cette image. Par exemple :
  1. `Plot 800 800` (lancement du programme)
  2. Clique gauche (saisie des sommets) + Clique milieu (fin saisie)
  3. Touche f (remplissage scan-line)

Je ne vous demande pas de me montrer toutes les fonctionnalités de votre programme. Il faut, dans un premier temps, que je puisse reproduire l'image que vous avez fournie en exemple sans avoir à éplucher le code. Si vous avez choisi d'ajouter des fonctionnalités vous aurez tout le loisir de m'en parler la semaine du 5 novembre. Mais je préfère vous informer que le projet sera évalué seulement sur les fonctionnalités spécifiquement demandées dans le sujet.