

Rapport de l'évaluation intermédiaire

Par Quentin Guardia, UPVD

Table des matières

Structures.....	2
Test unitaires.....	2
Commandes.....	3

Structures

Les points sont définis dans des structures. Chaque structure contient un flottant x et un flottant y , correspondant aux coordonnées du point concerné. Les nuages de points sont définis grâce à des tableaux de structures de points alloués dynamiquement. Il y a trois modules : point, nuage et geometrie2d. Chaque module a un fichier .c et un header. On peut choisir d'utiliser un des trois modules à partir du main, qui est un fichier à part.

Concernant le module point, la fonction point permet d'initialiser deux points que l'utilisateur définira au clavier. La fonction de comparaison prend deux points en paramètre et retourne 0 si le second est le plus grand, 1 si le premier est le plus grand, sinon 2. Il y a également une fonction afficher qui se contente d'afficher en sortie standard les coordonnées d'un point.

À propos du module nuage, un tableau de taille demandée à l'utilisateur est rempli au hasard grâce à la fonction hasard utilisant rand() pour obtenir des flottants compris entre 0 et 1. La fonction de tri par insertion permet de trier par ordre croissant les points du nuage de point, en réutilisant la fonction de comparaison des points du premier module. La fonction affichant les nuages est également basée sur celles affichant les points, sachant qu'un nuage est un tableau de points. On retrouve trois autres fonctions. La première enregistre les tableaux dans un fichier texte situé dans « sortie », la deuxième les lit, la troisième affiche les graphiques associés grâce à gnuplot et enregistre l'image associée dans le dossier « sortie ». La première ligne des fichiers textes comporte le nombre de points, il ne faut donc pas nécessairement connaître préalablement la taille du tableau enregistré dans un fichier pour lire ce dernier.

Le module geometrie2d permet quant à lui de trouver l'enveloppe convexe d'un nuage de point. Pour cela, deux fonctions majES et majEI empile respectivement sur leur pile les points de l'enveloppe supérieure et ceux de l'enveloppe inférieure, notamment grâce à la fonction orientation qui permet de déterminer si trois points appartiennent bien à l'enveloppe inférieure ou supérieure. La fonction env_convexe fusionne ces deux listes en conservant leur ordre afin de donner l'enveloppe convexe. Celle-ci est retournée et peut être affichée à l'écran via la fonction tracer, par l'intermédiaire de gnuplot une fois de plus. Cette fonction prend également en paramètre le nuage de points correspondant pour afficher les points et l'enveloppe sur un même graphique.

Les fichiers sont séparés en cinq dossiers : documentation, sources, objets, générés par l'utilisateur et réservés aux tests unitaires. Pour plus d'informations sur l'organisation des fichiers dans les dossiers, lire « readme.md », situé à la racine.

Test unitaires

Les tests unitaires sont succincts, mais l'exécution se déroule correctement. Les fonctions `tri_insertion`, `nuage`, `hasard`, `comparer_point`, `lire`, `enregistrer`, `orientation`, `majES`, `majEI` et `env_convexe` sont testées. La fonction `lecture` est d'abord testée grâce à un tableau initialisé. On compare la structure issue de la fonction `lire` avec la structure attendue. Ainsi on peut tester les autres fonctions en lisant le contenu de fichiers situés dans le dossier « test », ce qui permet de comparer les résultats obtenus aux valeurs attendues. Deux fonctions permettant de comparer, l'une les piles et l'autre les nuages de points, optimisent le code. Tout comme la fonction `res_test`, qui centralise l'affichage des succès et des échecs. Il faudrait tester avec plus de tableaux différents pour chaque fonction afin d'avoir une certitude plus élevée de la validité des tests, ce que j'ai commencé à faire grâce à la fonction testée `nuage` qui donne un nuage de points aléatoire.

Commandes

Pour générer l'exécutable, il faut taper « `make` ».

« `make clean` » nettoie les fichiers objets, les fichiers générés par l'utilisateur et l'exécutable.

« `./geometrie` » lance l'exécutable. Il faudra ensuite taper une lettre pour lancer au choix le programme ou bien les tests unitaires.