

Interfaces Graphiques

Projet 2015–2016 – Visualisation d’algorithmes sur les graphes

Le but du projet est d’écrire un programme permettant de présenter de façon dynamique l’exécution d’un algorithme sur les graphes. Le projet est en trois parties distinctes.

1 Éditeur de graphes

La première partie consiste à réaliser une interface graphique permettant d’éditer des graphes en utilisant la souris et le clavier.

L’utilisateur devra pouvoir construire son graphe sur l’écran. Il pourra choisir la forme des sommets – par exemple des cercles ou des rectangles, en choisissant éventuellement des formes différentes pour les sommets d’un même graphe. Il devra aussi pouvoir ajouter, placer et déplacer ces sommets à la souris, donner des noms aux sommets, et construire des arêtes (orientées ou non, en fonction de l’algorithme à implémenter). Bien sûr, il faut aussi donner la possibilité de supprimer des arêtes ou des sommets, et de modifier toute caractéristique d’un élément. L’éditeur devra aussi permettre d’effectuer sur le graphe ou une partie du graphe des transformations globales : au minimum des translations et de rotations.

Dans tous les algorithmes qui nous intéressent (voir la section suivante) on associe des poids aux arêtes (et parfois aussi aux sommets). L’éditeur de graphe devrait donner la possibilité de spécifier et de modifier ces poids. Les poids doivent être visibles à l’écran.

Si le graphe est dense sur l’écran, l’utilisateur peut avoir de doutes sur la correspondance entre les poids et les arêtes (par exemples si deux poids sont très proches à l’écran). Il faut donc prévoir un mécanisme permettant de vérifier quel poids correspond à quelle arête : par exemple, en cliquant sur une arête (ou en la sélectionnant de toute autre façon), le poids correspondant se mettra en surbrillance. Inversement, sélectionner un poids modifiera l’aspect visuel de l’arête associée.

Si les arêtes sont représentées uniquement par des lignes droites l’image du graphe risque d’être bien encombrée. Il serait donc judicieux de pouvoir construire des arêtes courbes. Toute autre idée rendant l’édition de graphe plus agréable pour l’utilisateur est la bienvenue : par exemple faire en sorte que, lorsque l’utilisateur déplace un sommet du graphe, les arêtes adjacentes restent « collées » au sommet déplacé, afin d’éviter d’avoir des arêtes qui pendent sans être attachées à leurs extrémités ; ou encore, donner la possibilité d’annuler une ou plusieurs opération d’édition.

Il est possible d’envisager une interface d’édition de graphes plus aboutie, avec des graphes en 3D, des sphères ou des cubes comme sommets, la possibilité d’effectuer une rotation du graphe dans l’espace, etc. Dans la version basique du projet, on ne demande pas des graphes en 3D, mais une version exploitant les possibilités que donne la troisième dimension peut rapporter des points supplémentaires.

2 Visualisation d’algorithmes sur les graphes

La deuxième partie du projet consiste à implémenter et visualiser le déroulement d’un algorithme (ou de plusieurs) sur un graphe. Le choix de l’algorithme est libre : plus court chemin (Dijkstra ou autre), flot maximal, arbre recouvrant de poids minimal, etc.

Le but de cette partie est de permettre à l’utilisateur de visualiser les changements d’états des sommets et/ou des arêtes ainsi que les données calculées pendant l’exécution de l’algorithme choisi : l’utilisateur doit pouvoir « voir » comment l’algorithme est exécuté étape par étape, par

exemple par des changements de couleur ou d'épaisseur de sommets ou d'arêtes. On peut imaginer le programme comme un outil didactique pour un cours d'Algorithmique.

Les états possibles dépendent de l'algorithme choisi : un sommet pourra par exemple être dans les états non visité, visité mais pas terminé, visité et définitivement terminé. L'algorithme peut aussi calculer des données intermédiaires ou finales, par exemple : une longueur de chemin ; les chemins les plus courts trouvés à un instant donné par l'algorithme de chemin le plus court ; une partie de l'arbre recouvrant pour l'algorithme de l'arbre recouvrant, etc.

Il y a au moins deux possibilités naturelles pour visualiser l'exécution :

- **L'exécution pas à pas.** Ici « pas » n'est pas à prendre au sens algorithmique mais au sens visuel. Faire un pas, c'est exécuter l'algorithme jusqu'au premier changement de la représentation visuelle du graphe : changement de couleur d'un sommet ou d'une arête, changement de l'étiquette d'un sommet ou d'une arête (si l'étiquette est une valeur calculée par l'algorithme).
- **L'exécution continue.** Les pas se déroulent de façon continue, à une vitesse réglable, par exemple à l'aide d'un Slider.

Les deux méthodes pourront être combinées : l'utilisateur pourra passer de l'une à l'autre et, lorsque l'exécution est continue, mettre en pause, reprendre, ou encore redémarrer l'exécution. Le contrôle de l'exécution pourra se faire au clavier, ou beaucoup mieux, à l'aide d'un boîtier de boutons.

Vous êtes libre de choisir la manière dont sont liés l'algorithme et sa visualisation. Il y a au moins deux possibilités (et même des possibilités médianes) :

- L'algorithme est synchronisé avec sa représentation visuelle, *ie.* la représentation visuelle montre l'état actuel de l'algorithme.
- L'algorithme est exécuté d'un seul coup, de façon invisible pour l'utilisateur. On stocke (en mémoire ou dans un fichier) tous les états et les valeurs intermédiaires. Un pas dans la représentation visuelle correspond au passage à l'état suivant dans les données intermédiaires.

Dans le second cas, l'interface graphique pourra permettre de visualiser l'exécution de l'algorithme en sens inverse, de rembobiner l'exécution, de la reprendre, de la commencer à une étape intermédiaire, etc.

3 Sauvegarde du graphe dans un fichier

Dans l'éditeur de graphe, on ajoutera la possibilité de sauvegarder le graphe en cours de construction dans un fichier. Vous êtes libre de choisir le format de fichier, mais le format xml est bien commode pour stocker les informations structurées. L'utilisateur pourra éditer un graphe, le sauvegarder, puis relire le fichier soit pour continuer l'édition du graphe, soit pour lancer l'exécution de l'algorithme.

Cette partie du projet est primordiale pour que votre projet soit utile en pratique, mais elle est moins pertinente pour un cours d'Interfaces Graphiques, donc elle n'est à faire que si le temps le permet – il vaut mieux une interface graphique riche et conviviale sans sauvegarde, qu'une sauvegarde avec une interface médiocre. Cela dit, pour la démonstration de votre projet pendant les soutenances, il serait bien utile de préparer à l'avance au moins un graphe de taille conséquente (une dizaine de sommets), pour montrer de façon convaincante comment marche la visualisation.

4 Modalités

Le projet est à réaliser en binôme (ou en monôme). Les équipes de plus de deux personnes **ne sont pas autorisées**, sans aucune exception.

Les soutenances auront lieu début de mai. Les dates et les modalités de rendu seront précisées ultérieurement.