

Mise en page d'un paragraphe de texte

Sujet proposé par Gilles Schaeffer

`Gilles.Schaeffer@lix.polytechnique.fr`

URL de suivi : <http://www.enseignement.polytechnique.fr/profs/informatique/Gilles.Schaeffer/INF421/projetX13.html>

1 Présentation du sujet

Dans un éditeur de texte tel que Microsoft Word ou son équivalent chez OpenOffice.org la mise en page d'un paragraphe de texte est effectuée automatiquement pour peu que l'utilisateur n'insère pas de retour à la ligne explicite. Dans ces outils, le placement des mots dans les lignes est effectué de manière gloutonne : tant que les mots rentrent dans la ligne courante on les ajoute, et dès que ce n'est plus le cas on passe à la ligne suivante.

Par défaut l'espacement entre les mots est constant et la longueur des lignes variable ce qui provoque un effet de dentelle peu agréable. Pour éviter ce désagrément ces éditeurs proposent l'option de "justifier le texte", c'est à dire d'aligner l'extrémité des lignes à gauche et à droite, quitte à utiliser un espacement entre mots différent pour chaque ligne. Cependant cette méthode de mise en page conduit fréquemment à avoir des mots trop espacés au sein d'une ligne. L'effet est encore plus désagréable lorsqu'une telle ligne "trop aérée" est encadrée par deux lignes faites de petits mots rapprochés : on serait tenté de déplacer un petit mot ou deux vers la ligne centrale pour "équilibrer" l'espacement, ce qu'on peut faire en utilisant des retours à la ligne forcés (insérés en général avec la combinaison de touches "shift+enter" car la touche "enter" termine par défaut un paragraphe), mais ce genre d'ajustement manuel est long à effectuer et ne peut être fait qu'une fois que le texte est définitivement fixé (ou bien doit être refait à chaque modification du texte).

A contrario, d'autres logiciels de traitement de texte, comme le logiciel TeX/LaTeX, utilisent une méthode de calcul plus sophistiquée, basée sur la programmation dynamique pour choisir l'endroit où insérer les retours à la ligne, de façon à minimiser l'espacement maximal utilisé entre deux mots (ou alternativement, minimiser l'effet de dentelle dans le cas des textes justifiés à gauche seulement).

L'objectif de ce projet sera de proposer un outil pour optimiser *a posteriori* la mise en page d'un texte écrit avec Microsoft word ou Openoffice.org : une fois le texte terminé (prêt à être optimisé), l'utilisateur le sauvegarde au format xml (format qui permet de retrouver facilement les informations telles que taille des pages/marges et polices de caractères utilisées)

et lui applique l’outil d’optimisation pour insérer des retours à la ligne forcés aux positions optimales dans chaque paragraphe. L’utilisateur peut alors recharger le texte dans son éditeur pour voir le résultat et éventuellement le sauvegarder à un format propriétaire plus classique (.doc ou .odt) ou l’exporter en pdf pour le figer et/ou l’imprimer.

2 Présentation de l’algorithme

La version finale du sujet donnera ici une description informelle de l’algorithme de calcul. Il s’agit d’une application directe des idées de programmation dynamique vues en cours.

3 Détails techniques

La version finale du sujet fournira ici quelques indications pour aider à l’interprétation d’un fichier xml et ainsi qu’au sujet du calcul de la longueur des mots en fonction de la police de caractère utilisée.

4 Travail à effectuer

4.1 Objectif principal

Le but du projet est de fournir un exécutable qui prend un fichier xml en entrée, lui applique l’algorithme de mise en page, et refabrique un fichier xml en sortie.

Au minimum le programme doit pouvoir traiter un unique paragraphe utilisant une police de caractères de taille fixée, mais il serait raisonnable de savoir traiter le cas de plusieurs paragraphes utilisant chacun sa police de caractères. *A priori* on ne demande pas de traiter le cas d’un paragraphe mélangeant plusieurs polices de caractères.

4.2 Extensions possibles

- Une première extension possible est d’autoriser de couper les mots (en respectant les règles à ce sujet). Couper trop de mots rend le texte peu lisible, il faut donc prévoir un malus lié à l’utilisation de cette possibilité, adapter l’algorithme en conséquence et faire des tests pour calibrer le malus appliqué.
- On pourrait vouloir utiliser le même type d’approche pour placer les lignes dans les pages en évitant de couper des paragraphes trop près de la fin, et surtout pour choisir le placement des éventuelles figures “flottantes” : à quel endroit les placer proche de l’“ancrage” proposé par l’utilisateur, mais de façon à ce que chaque page contienne une quantité raisonnable de texte.