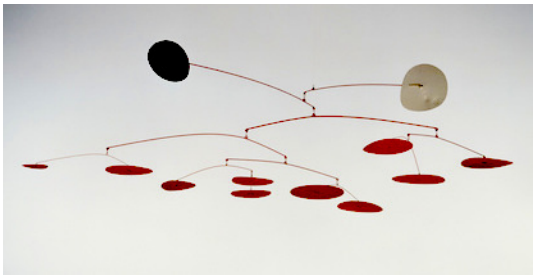
Construction de mobiles

Projet LS4

2014



1 Description du projet

Le but du projet est de réaliser une application avec affichage graphique qui construit des mobiles équilibrés à partir d'une liste de poids donnés en entrée.
Ci-contre : Mobile d'Alexander Calder intitulé Black, White and Ten Red.
Source : `http://calder.org/work/by-category/hanging-mobile`

2 Modalités

Les projets sont à faire par groupes de deux étudiants. Une fois les groupes constitués, vous devrez vous inscrire sur DidEL afin que nous puissions mettre en place les soutenances.

Votre code devra être bien structuré et commenté. Il doit pouvoir tourner sur les machines de la fac sans modification. Aucun module python en-dehors des modules standard ne doivent être utilisés. (En cas de doute, consultez votre enseignant.) Si vous travaillez sur votre machine personnelle, c'est à vous de tester le bon fonctionnement de votre programme sur les machines de la fac avant de le rendre sur Didel. Le code déposé sur Didel sera celui qui sera utilisé lors de la soutenance.

Vous devrez soumettre un rapport avec votre projet. Ce rapport n'a pas besoin d'être très long mais il doit comporter les éléments suivants :

- Petit manuel d'installation et d'utilisation. Donnez les instructions détaillées qui permettent de voir tourner votre application.
- Liste des fonctionnalités implémentées.
- Liste des hypothèses retenues, le cas échéant.
- Liste des comportements erronés ou inattendus.
- Liste des fonctionnalités non-implémentées.
- Description des principaux algorithmes utilisés.
- Quelques exemples illustrant les algorithmes implémentés, avec des explications.
- Bilan du travail réalisé, améliorations possibles, etc.

Les soutenances du projet consisteront en une présentation du programme par le groupe, puis par la démonstration de votre application suivi de questions. Chacun devra être capable de répondre aux questions concernant la partie réalisée par l'autre. Les notes seront individuelles. La préparation de la soutenance est indispensable. Les fichiers servant à la démonstration doivent être préparés à l'avance et doivent permettre de voir l'ensemble du travail, tout en respectant le temps imparti. Des tests qui illustrent les limites de votre travail sont appréciés car ils démontrent que vous avez pensé à tester tous les cas.

Dates

13 avril : Inscription des groupes sur DidEL.

29 avril 17h : Rendu du projet sur DidEL. *Tout retard sera pénalisé, et aucun projet ne sera accepté après 18h.* Il est vivement conseillé de prévoir le temps nécessaire en cas d'inattendus ou de de surcharge au dernier moment. En cas de réel empêchement, prenez contact avec votre enseignant *avant* l'échéance.

5-6-7 mai : Les soutenances de projet se dérouleront pendant cette période et la date sera précisée ultérieurement.

Notation

Le projet sera noté suivant le barême indicatif suivant :
- sur 12 points : toutes les fonctionnalités obligatoires
- sur 8 points : Les fonctionnalités supplémentaires et l'originalité du projet.

La note porte sur la réalisation du projet, ainsi que sur le rapport et la soutenance.

3 La physique de l'équilibre

Vos mobiles doivent respecter les lois de la physique. Si on applique un poids p_1 du côté gauche d'une tige de longueur ℓ et un poids p_2 du côté droit, alors il faut, pour que les deux poids s'équilibrent, tenir la tige à une distance ℓ_1 de l'extrémité gauche, de sorte que $p_1\ell_1 = p_2(\ell - \ell_1)$. Si on résoud cette équation pour ℓ_1, on a $\ell_1 = \frac{p_2\ell}{p_1+p2}$. En physique, cela correspond à rendre égaux les *moments de torsion* des deux côtés. Cette description est une simplification de la loi des moments de torsion, et de façon plus générale, on doit tenir compte de l'angle avec lequel est appliquée la force. Ici nous ne considérons que le cas où l'angle est toujours perpendiculaire au centre de gravité.

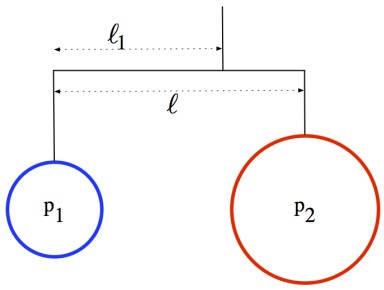


FIGURE 1 – Pour un poids p_1 du côté gauche d'une tige de longueur ℓ et un poids p_2 du côté droit, l'équilibre se trouve au point $\ell_1 = \frac{p_2\ell}{p_1+p2}$.

Vos mobiles seront constitués de n objets de poids $p_1, \ldots, p_n$ donnés en entrée dans un fichier. Les objets sont suspendus par des fils à chaque extrémité des tiges. La longueur des fils ou des tiges n'est pas importante. Vous pouvez les choisir selon vos préférences.

Une collection d'objets peut être disposée de plusieurs façons différentes pour former des mobiles différents. **Vous devez implémenter au moins un algorithme qui dispose les objets afin de former un mobile.** Par exemple, un algorithme pourrait consister à placer la moitié des objets à gauche, l'autre moitié à droite, puis procéder récursivement pour placer les éléments dans les sous-mobiles de gauche et de droite. Vous pouvez imaginer d'autres stratégies pour placer les objets pour obtenir des résultats différents, par exemple prendre une proportion d'objets différente à gauche et à droite, trier ou non les poids, procéder au hasard, etc.

Pour un affichage simple, vous pouvez supposer qu'un objet de poids p est un cercle de rayon proportionnel à $\sqrt{p}$, ou une sphère de rayon proportionnel à $\sqrt[3]{p}$. Vous pouvez négliger le poids des tiges et des fils.

4 L'arbre du mobile

La structure de données utilisée pour représenter un mobile est un arbre binaire. Chaque noeud représente soit une tige sur laquelle sont suspendus deux sous-mobiles, soit un objet à accrocher. Si c'est un objet à accrocher, le noeud contient une valeur qui représente le poids de l'objet. Si c'est un sous-mobile, alors le noeud contient deux sous-arbres. Le sous-arbre de gauche représente le sous-mobile accroché à l'extrémité gauche de la tige, le sous-arbre de droite représente le sous-mobile accroché à l'extrémité droite de la tige,

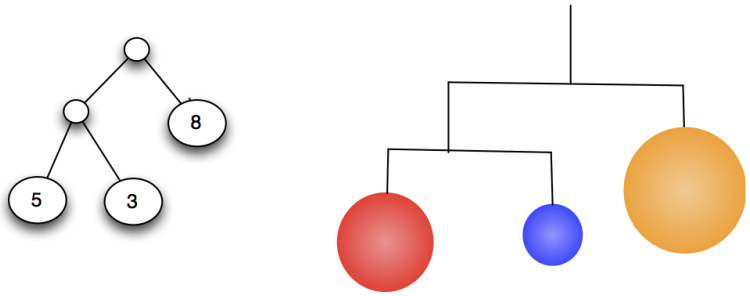


FIGURE 2 – À gauche, l'arbre représenté par la chaîne `[[5,3],8]`. À droite, le mobile correspondant à cet arbre.

5 Les fonctionnalités obligatoires

Votre programme doit pouvoir fonctionner suivant deux modes :

1. L'arbre des poids est décrit dans un fichier et vous affichez le mobile correspondant. Dans ce cas vous devez construire l'arbre demandé à partir du fichier, puis l'afficher.
2. Seuls les poids sont donnés dans un fichier. Dans ce cas, votre programme doit d'abord construire l'arbre en suivant l'algorithme de votre choix, puis l'afficher, et enfin écrire l'arbre dans un fichier suivant le format demandé.

Dans les deux cas, la même fonction d'affichage doit être utilisée.

6 Le format des fichiers de données

Liste de poids. La liste est donnée dans un fichier avec un poids par ligne. La dernière ligne du fichier est vide.

Arbre. Vous devez pouvoir lire un fichier qui contient un arbre et l'afficher, mais aussi écrire dans un fichier l'arbre que vous avez construit à partir de poids. On peut représenter un nœud d'un arbre binaire par un tableau à deux éléments. Le premier élément du tableau représente le sous arbre gauche et le second élément représente le sous-arbre droit. Ainsi l'expression `[[5,3],8]` représente l'arbre donné dans la figure 2. Vous devez utiliser ce format en lecture ainsi qu'en écriture.

Note : Consultez la fonction `eval()` de python qui prend en entrée une chaîne et retourne l'objet python représenté par cette chaîne. Par exemple,

```
s = '[2,3,7]' # s est une chaine de caracteres
l = eval(s)   # l est un tableau contenant les trois elements
```

7 Les fonctionnalités supplémentaires

Vous devez personnaliser votre projet en ajoutant des fonctionnalités supplémentaires. Voici quelques suggestions. Des points seront donnés pour l'originalité.

1. Les mobiles ne sont pas forcément équilibrés, c'est-à dire que les tiges ne sont pas nécessairement horizontales. Dans ce cas, il faut tenir compte des angles dans le calcul des moments de torsion.
2. Implémenter plusieurs algorithmes différents pour la construction du mobile à partir de la liste des poids.
3. Générer automatiquement des listes de poids qui donnent lieu à des mobiles intéressants.