

CONSTRUCTION DE MOBILES

INSTALLATION

L'application ne nécessite aucune installation à part de changer la première ligne du fichier mobile.py afin de pouvoir l'exécuter avec sa propre version de python. Un dossier « res », contenant les ressources nécessaires à l'application, doit être également présent.

UTILISATION

L'interface graphique propose différentes options de génération de mobile et d'affichage.

- Une barre de menu donne accès à toutes les options de l'application : ouvrir/sauvegarder un mobile, algorithme de génération de mobile et méthode d'affichage.
- Une barre de raccourcis donne accès rapidement aux fonctions principales.
- Une barre de statuts indique quel fichier est actuellement chargé.

FONCTIONNALITES

Fichiers :

- Lecture de deux formats : un arbre au format d'un tableau écrit sur une seule ligne (ex : [2,[[4 ,5]]) ou une liste de poids (un poids par ligne et terminant par une ligne vide).
- Sauvegarde sous les deux formats décrits précédemment.

Génération de mobile :

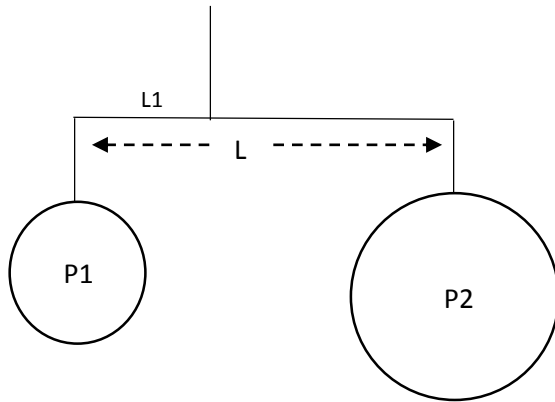
Toutes les générations se font à partir d'une liste de poids.

- Simple : la liste de poids est parcourue dans l'ordre. Pour chaque poids, si son poids est plus élevé que l'arbre, on le place à la racine, sinon on le descend récursivement du côté le plus léger. On obtient ainsi un mobile pas trop déséquilibré et d'une profondeur faible.
- Équilibrée : on teste toutes les combinaisons possibles de la liste de poids de façon à ce que le total des poids de cette combinaison soit égale à la moitié du poids total de la liste (on se donne une marge d'erreur k). Dès qu'une combinaison respectant cette condition est trouvée, la liste est divisée en deux sous-listes contenant la combinaison pour l'une et le reste des poids pour l'autre. Ces deux listes donneront les deux sous-arbres du nœud courant et on réapplique l'algorithme sur les sous-listes. Dès que l'on trouve un arbre respectant cette condition pour chaque nœud, l'algorithme se termine. Si jamais aucun arbre n'est trouvé pour la marge d'erreur donnée, on incrémente la marge d'erreur et réapplique l'algorithme.
- Maximum droit/gauche, Minimum droit/gauche : ces arbres respectent la règle suivante : à chaque nœud, le nœud gauche/droit est le maximum/minimum du sous-arbre. Pour obtenir simplement ce mobile, la liste de poids est triée dans l'ordre croissant (maximum) ou décroissant (minimum) puis les poids sont ajoutés au mobile dans l'ordre de la liste suivant la règle droite/gauche.

Affichage :

Les coordonnées des nœuds sont calculées indépendamment des autres de façon à ce que l'origine se trouve sur le point d'accroche du nœud. Ensuite le mobile est dessiné sur la zone graphique et pour finir il est mis à l'échelle de la zone graphique afin de la remplir.

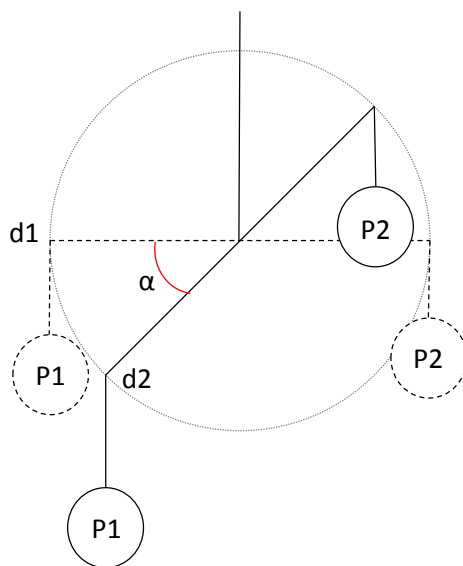
- Physique simple : le calcul des coordonnées respecte les moments de torsion simple (dans tenir compte des angles)



Pour qu'il y soit équilibré, la formule suivante doit être respectée : $p1 \times l1 = p2 \times (l - l1)$

Figure 1 - Physique simple

- Physique avec rotation : ici, les deux sous mobiles d'un nœud se situent à équidistance l'un de l'autre. La droite L subit une rotation α selon le poids des sous mobiles.



Il faut commencer par trouver la valeur de l'angle de rotation :

$$\alpha = \sin^{-1}\left(\frac{(p1-p2)}{\text{rayon}}\right)$$

Les coordonnées de d2 se calcul simplement ensuite :

$$d2(x,y) = ((-\text{rayon}) \times \cos(\alpha), \text{rayon} \times \sin(\alpha))$$

Figure 2 - Physique avec rotation

EXEMPLES :

Algorithme simple :

Prenons la liste (4,1,3,4,2) et plaçons déjà les deux premiers poids de la liste sur le même nœud: on obtient donc [4,1]. Puis on place les suivant un à un dans l'ordre. Pour 3, le côté droit est plus léger donc on met le 3 de ce côté pour obtenir [4,[1,3]]. Pour le 4, les deux côtés sont de même poids, donc on le met sur le côté gauche par défaut pour obtenir [[4,4] , [1,3]]. Pour finir le 2, le côté droit est le plus léger, donc on regarde sur le sous nœud [1,3]. Ici le côté gauche est le plus léger donc on obtient [[4,4], [[1,2], 3]].

Algorithme équilibré :

Prenons la liste (4,1,3,4,2), son total est de 14 et cherchons un mobile avec une marge d'erreur de 1. Cherchons d'abord les combinaisons égales à 7 ± 1 . Il est évident que celle de cardinal 1 ne marchent pas, cherchons celle de cardinal 2 en prenant la première valeur de la liste et en testant avec les autres une à une : $(4,1) = 5$ non, $(4,3) = 7$ oui ! On fait donc 2 sous listes (4,3) et (1,4,2) qui ont toutes les deux leur total égale à 7 ± 1 .

On réapplique le même raisonnement : (4,3) a un total de 7 donc les sous listes doivent être 3 ± 1 ce qui marche évidemment ici. Pour l'autre, son total est de 7, on commence par 1 : $(1,4) = 5$ non, puis $(1,2) = 3$ oui. On fait donc deux sous listes (1,2) et (4) et on réapplique une dernière fois le raisonnement sur (1,2) qui est trivial.

On obtient donc le mobile [[4,3] , [[1,2] , 4]] qui est équilibré avec une marge d'erreur de 1.

COMPORTEMENTS INATTENDUS / ERRONES

La mise à l'échelle du mobile lors de l'affichage ne fonctionne pas toujours parfaitement. Il arrive que les poids les plus bas soient rognés. De plus la remise à l'échelle ne se fait pas automatiquement en cas de redimensionnement de la fenêtre : l'évènement déclenché pendant le redimensionnement n'a pas été trouvé.

L'affichage des mobiles générés par les algorithmes de minimum/maximum gauche/droite par la physique avec rotation n'est pas très intéressant visuellement.

BILAN

Toutes les fonctionnalités demandées ont été implémentées hormis la génération de listes de poids intéressantes car nous avons préféré chercher un algorithme performant de recherche du mobile le plus équilibré sur une liste de poids donnée et avoir une interface graphique la plus intuitive possible avec le moins d'erreurs ou bugs durant l'utilisation de l'application.

De plus, la structure d'arbre implémentée pour le mobile est facilement modulable, surtout au niveau de son affichage car il ne tient pas compte de la partie graphique. Il suffit simplement de mettre les coordonnées de chaque partie de l'arbre à l'échelle de la zone graphique où l'on souhaite afficher notre mobile (les coordonnées de chaque poids/nœud étant calculées avec pour point d'origine le nœud/poids lui-même).