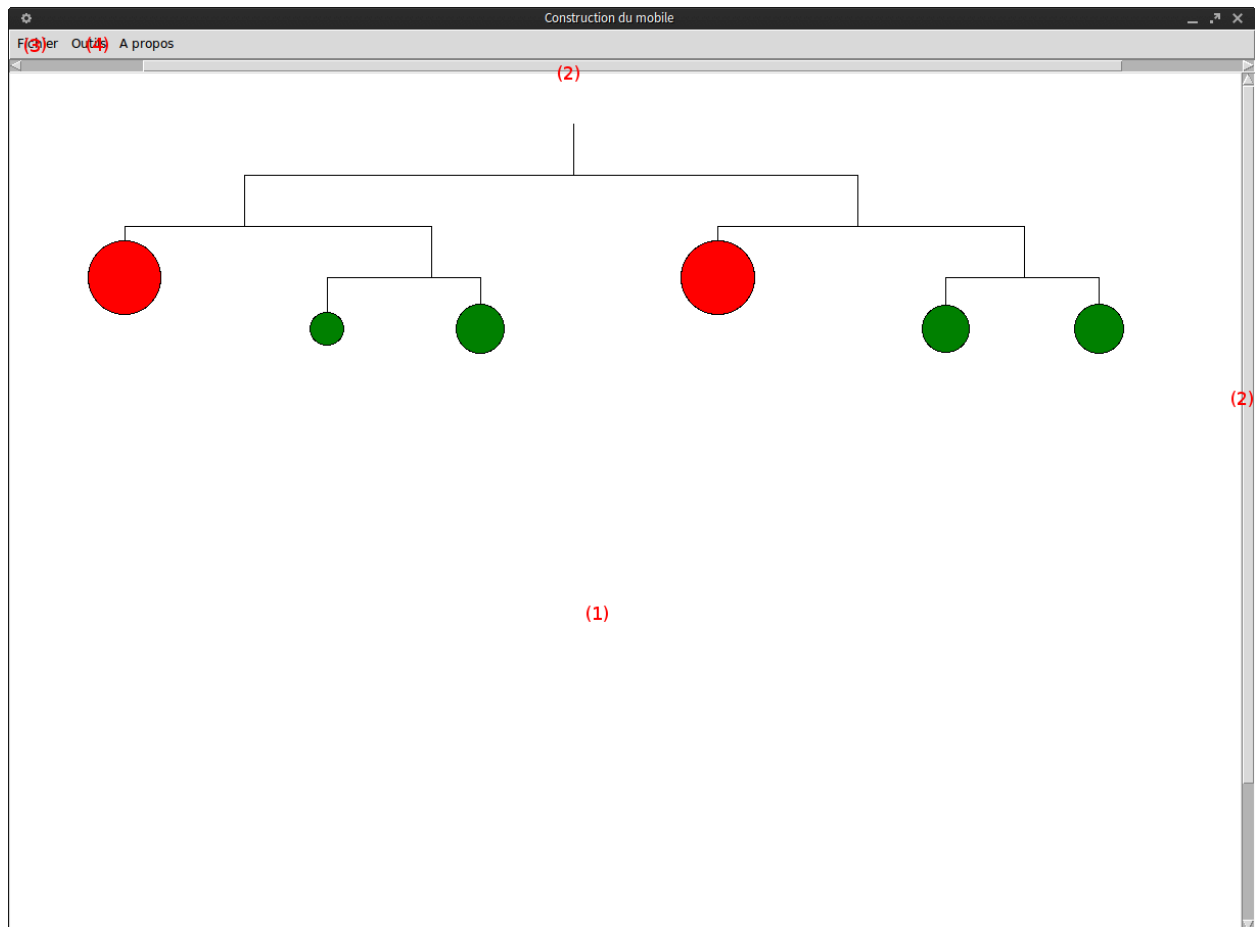


Projet LS4

<< Construction de Mobiles >>

Jérôme BETTINELLI(Info 3) - Thomas HAUTIER(Info 2)

Mode d'emploi



(1) Zone d'affichage

(2) Scroll(Horizontal et Vertical)

(3) Fichier :

- **Ouvrir** : Permet d'ouvrir un fichier représentant un mobile. Détection automatique du type de fichier.
- **Enregistrer** : Permet d'enregistrer le fichier
- **Quitter** : Permet de quitter le programme

(4) Outils :

- **Aléatoire** : Permet de générer aléatoirement un mobile.
- **Modifier les valeurs** : Permet de modifier les poids de l'arbre et de régénérer l'arbre. Une légende est présente pour aider à identifier les poids.

Le projet se lance avec la commande `python Projet.py`

Modes

A chaque mode correspond un algorithme :

Mode 1 :

Le premier mode est une simple répartition récursive de la liste de poids en décomposant la liste en 2 parties jusqu'à obtenir une liste affichable.

Mode 2 :

Dans un premier temps la liste de poids est découpée en deux liste qu'on remplit alternativement après avoir triée la liste d'origine.

On affiche ensuite pour chaque fils de la racine un arbre où le poids le plus lourd est affecté au fils gauche et tout le reste au fils droit récursivement .

Mode 3 :

On reprend le même principe que le second mode sauf que l'on alterne la place du poids le plus lourd entre gauche et droite à chaque niveau de l'arbre.

Mode 4 :

On essaie de faire un arbre le plus équilibré possible au niveau des poids.

On commence par trier la liste par ordre croissant.

On découpe ensuite la liste récursivement en deux sous listes qu'on va essayer d'équilibrer en ajoutant dans un premier temps le poids le plus important et le second dans chacune des listes.

On essaie ensuite de les équilibrer en affectant chaque élément après une comparaison de la somme des deux listes.

Organisation du projet

Le projet est organisé en 6 fichiers sources et 4 fichiers d'exemples.

Fichiers sources :

Arbre.py : Représente la structure d'un arbre. Et contient des fonctions (souvent récursives) qui seront appliquées à l'arbre.

Fenetre.py : Contient toute la partie graphique du projet (traitement des fichiers inclus).

Noeud.py : Représente la structure d'un noeud, avec un fils gauche ou un fils droite. Contient des fonctions qui seront exécutées selon si ses fils sont des noeuds ou des poids ainsi que des fonctions d'affichage.

Mobile.py : Représente la structure d'un poids. Contient des fonctions d'affichages.

Projet.py : Contient un code pour créer une instance de fenêtre.

Fichiers d'exemples:

test1 : Contient un arbre déjà organisé.

test2 : Contient un autre arbre déjà organisé mais plus grand.

test3 : Contient une liste de poids (l'organisation est donc laissée à l'utilisateur)

test4 : Contient une liste de poids plus importante.