

TP

Pattern MVC - The Interactive Camembert

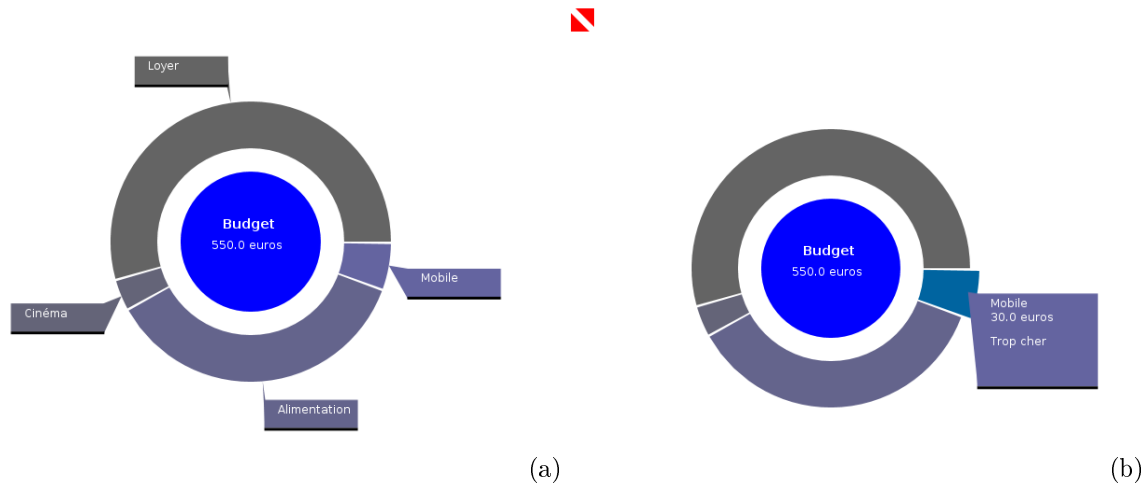


Figure 1: The Interactive Camembert: (a) un composant graphique interactif pour afficher des camemberts à partir d'un modèle de données; (b) lorsque l'utilisateur clique sur un quartier, celui-ci est mis en valeur et des informations complémentaires peuvent être affichées. A l'aide des flèches de navigation représentées en haut sur le camembert (b), l'utilisateur navigue vers le quartier suivant ou précédent.

L'objectif de ce TP est d'utiliser un composant graphique pour l'affichage de camemberts interactifs (tel qu'illustré dans la Figure 1) en suivant le design pattern MVC (Model-View-Controller) qui sera étendu avec des **interfaces** de **contrôleur**, de **modèle** et de **vue** pour une meilleure modularité. Il faut proposer une implémentation du **modèle** (qui contiendra les données), une implémentation de la **vue** permettant la représentation des données du **modèle**, et un **contrôleur** permettant d'interagir sur ce camembert. Les fonctionnalités à implémenter sont:

- coté modèle: une structure de données qui stocke le titre (qui décrit le graphique), et un ensemble d'items. Un item est représenté par un intitulé (eg "Loyer" dans la figure ci-dessus), une description textuelle, et une valeur numérique.
- coté vue:
 - affichage au centre du camembert du titre et du total des valeurs numériques des items (il faudra donc "adapter" le modèle).
 - affichage d'un quartier de camembert par item (la taille du quartier est un ratio de la valeur numérique de l'item sur le total des valeurs), ainsi qu'une description textuelle (schéma de gauche sur la figure)
 - lorsque l'utilisateur clique sur un quartier, le quartier est mis en valeur et sont affichées la valeur numérique associée au quartier ainsi que la description textuelle. De plus deux boutons peuvent être ajoutés pour naviguer vers le quartier suivant ou précédent (utile si certains quartiers sont difficiles à sélectionner car trop petits). Cette représentation correspond au schéma de droite sur la figure.

- lorsque l'utilisateur clique en dehors d'un quartier, le quartier est désélectionné.

Exercice 1.1 (Implémentation du Modèle)

1. Dans un premier temps, proposez une structure de données pour représenter les données du modèle. Ce modèle doit être indépendant de toute représentation graphique, et ne doit pas contenir de code de notification des changements du modèle (eg envers la vue)
2. Ajoutez des accesseurs et des modificateurs sur les items ainsi que deux méthodes pour ajouter ou supprimer des items.
3. Utilisez ensuite un design pattern Adapteur pour intégrer un système de notification des changements du modèle vers la/les vues (par exemple via des observateurs).

Exercice 1.2 (Implémentation de la Vue)

1. Implémentez ensuite un composant graphique à l'aide des squelettes Java/Swing fournis pour afficher un camembert à partir des données du modèle.

Exercice 1.3 (Implémentation du Controleur)

1. Commencez par ajouter un `MouseListener` à votre composant graphique. Pour déterminer la primitive géométrique qui a été cliquée par l'utilisateur, les classes `Arc2D` et `Rectangle2D` offrent une méthode `contains(double x, double y)` qui retourne si le point de coordonnée (x,y) est dans la primitive. Une fois le composant cliqué déterminé, mettez-le en valeur et affichez les données correspondantes.
2. Lorsqu'une primitive est sélectionnée, ajoutez deux boutons mettre en valeur respectivement la donnée suivante ou précédente dans le camembert.

Exercice 1.4 (Extra)

1. Pour finir, interfacez votre modèle de données avec la classe abstraite `TableModel` (`javax.swing.table`), de façon ensuite à pouvoir afficher une `JTable` en même temps que le camembert, et d'interagir avec cette `JTable`. Par exemples une sélection d'un item sur la `JTable` provoque la selection du même item dans le camembert (et inversement).