**Evaluation iOS**LP Informatique Répartie et Mobile, 2015  
[*guillaume.chiron@univ-lr.fr*](mailto:guillaume.chiron@univ-lr.fr)- [*michel.mernard@univ-lr.fr*](michel.mernard@univ-lr.fr)

**Application Congélateur :**

Cette application reprend les mêmes concepts que ceux abordés dans l'étape 1, 2 et 4 du mini-projet (ainsi que certaines notions vues en TP). Vous pouvez vous inspirer des travaux que vous avez déjà réalisés.  
  
Le but de cette nouvelle application est d'afficher dans un **tableView** une liste de catégories de produits.  
Lors d'un clique sur une catégorie, une seconde vue avec un autre **tableView** apparaitra pour montrer les liste des produits correspondant à cette catégorie.  
La liste des catégories/produits vous est donnée sous la forme d'un fichier JSON (fichier fourni avec cet énoncé).

**Au minium :**

- Créer une application de type "Master-Detail Application"

- A la fin du chargement de la vue **Master** (viewDidLoad), charger les catégories de produit dans le **tableView.** Ci-dessous le code permettant de charger un fichier JSON, de le parser et de parcourir les données organisées dans un dictionnaire. Ce code est volontairement non commenté, il serait peut-être intéressant de le commenter et d'en parler lors de la soutenance...

    NSData \*congelateurData = [NSData dataWithContentsOfFile:@"/Users/.../ congelateur.json"];

    NSError \* error;

    NSDictionary \* categorieDictionary = [NSJSONSerialization JSONObjectWithData: congelateurData options:kNilOptions error:&error];

    for (NSString\* keyCategorie in categorieDictionary) {

       NSDictionary \* **produitDictonary** = [congelateurDictionary objectForKey: keyCategorie];

for (NSString\* keyProduit in **produitDictonary**) {

NSDictionary \* **detailDictonary** = [**produitDictonary** objectForKey: keyProduit];

         int qte = [**detailDictonary** objectForKey:@"Qte"];

        NSLog(@"Quantité de %@ (%@) = %d", keyProduit, keyCategorie, qte);

}

    }

- La vue **Détail** apparait lors du clique sur un magasin; Ajouter une carte et centrer celle-ci sur les coordonnées du magasin sélectionné. Ajouter aussi une punaise pour afficher des informations supplémentaires.

- Sur la vue **Détail**, ajouter un **webView** en parallèle de la carte afin d'afficher des informations complémentaires (ou images) sur le magasin en question (à partir de l'url indiqué dans le fichier JSON)...

**Pour aller plus loin (pour ceux qui le souhaitent) :**

- Sur la carte, afficher à l'aide de punaises tous les magasins concurrents de la même ville que le magasin sélectionné.

- Pour ceux qui ont réussi à faire marcher le parsing HTML via Wikipedia lors du mini-projet, vous pouvez aussi afficher une punaise au centre de la ville d'appartenance du magasin de manière à apprécier la distance entre ce dernier et le centre ville.

--- **BON COURRAGE ---**