# **Énoncé du TD 1**

### **Objectif du TP**

L'objectif de ce TP est de développer une application mobile (très simplifiée) de sensibilisation au tri sélectif des déchets. Elle est de loin inspirée par [un article paru sur le Huffington Post](https://www.huffingtonpost.fr/life/article/perdu-dans-le-tri-selectif-on-decrypte-tous-les-pictogrammes-de-vos-produits_187752.html) le 17 octobre 2021.

#### **Création de l'écran d'accueil**

Créez l'écran principal de l'application avec un label et quatre boutons. Modifiez le texte du label et des boutons selon la figure ci-dessus (les images des boutons seront personnalisées plus bas).

#### **Création des quatre écrans secondaires**

Créez quatre contrôleurs de vue secondaires, et personnalisez leur couleur d'arrière-plan.

Créez des transitions (segue) entre l'écran d'accueil et chacun des quatre écrans secondaires.

#### **Intégration d'un NavigationController**

Placez l'écran d'accueil (Embed In...) sous le contrôle d'un contrôleur de navigation, et donnez un titre aux *navigation items* des cinq contrôleurs. Vérifiez que les boutons de navigation (Back) des écrans secondaires, prennent le nom de l'écran d'accueil.

#### **Importation des images**

Téléchargez les images contenues dans l'archive [ImagesTri.zip](https://moodle.univ-lyon2.fr/pluginfile.php/522461/mod_page/content/12/ImagesTri.zip), et importez-les dans les "assets" du projet Xcode.

Définissez les images des trois premiers boutons de l'écran d'accueil (champ "Image" dans l'inspecteur de propriété du bouton), à l'aide des fichiers image portant le suffixe "\_b" pour "button".

Modifiez le placement de l'image du deuxième bouton (champ "Placement" à "Trailing" dans l'inspecteur de propriété du bouton).

Placez les trois images portant le suffixe "\_f" pour "fullscreen" dans les trois premières vues secondaires "déchets électroniques", "déchets plastiques" et "déchets organiques", et redimensionnez-les à la taille des écrans.

Placez et dimensionnez les trois images portant le suffixe "\_p" pour "poubelle" en bas de la quatrième vue secondaire "jeu du tri". Définissez la propriété "Highlighted" de chacune des trois UIImageView, avec l'image portant le suffixe "\_h" pour "Highlighted". Vérifiez que l'aspect de la poubelle change lorsque vous cochez ou décochez la case à cocher State --> Highlighted de l'inspecteur de propriétés.

Placez à des positions arbitraires de la quatrième vue, les six images d'objets à recycler, et redimensionnez-les.

### **Programmation de l'écran de jeu**

La suite du TP va être consacrée à la programmation du 4ème écran secondaire intitulé "Jeu du Tri". Il s'agit de concevoir une application de glisser-déplacer, dans laquelle le joueur est appelé à déposer chaque objet dans le bac de tri approprié. S'il dépose l'objet en dehors d'une poubelle, ou sur le mauvais container, l'objet doit revenir à sa position initiale. Si le joueur a choisi le bon bac de tri, l'objet doit disparaître. Lors du survol d'une poubelle avec un objet, la poubelle passe en état "Highlighted".

#### **Création du fichier swift**

Créez une nouvelle classe "Cocoa Touch" nommée "JeuViewController", héritant de la classe UIViewController (pour cela, dans le menu File ... New .. File ... choisissez "Cocoa Touch Class", nommez-le), et ajoutez le fichier swift aux fichiers du projet.

Dans l'inspecteur d'identité de l'écran de jeu, changez la classe par défaut (UIViewController) en précisant la classe nouvellement créée "JeuViewController".

#### **Création des Outlet Collections**

Créez une "Outlet Collection" nommée "poubelles" et une "Outlet Collection" nommée "dechets". Reliez chacune des poubelles et chacun des déchets de la vue à l'Outlet Collection qui lui correspond. Pour faciliter l'affectation des déchets à son bac, respectez l'ordre des poubelles et l'ordre des déchets. Souvenez-vous qu'il y a un bug à la création des Outlet Collections, et qu'une fois l'Outlet Collection créée, il faut passer par l'inspecteur de connexions pour établir la connexion entre l'Outlet Collexion déjà créée et un nouvel objet.

#### **Déplacement des objets (drag)**

Pour rendre un objet déplaçable, il faut surcharger les méthodes touchesBegan et touchesMoved du JeuViewController. Lors du premier contact (touchesBegan) d'un doigt avec l'écran, il faut tester chaque objet pour voir s'il contient les coordonnées du doigt. On peut utiliser la méthode func contains(CGPoint) -> Bool de la classe CGRect, qui renvoie vrai si le rectangle auquel on applique cette fonction contient le point qu'on lui transmet. Pour éviter de déplacer plusieurs objets simultanément, il faut mémoriser l'objet sur lequel touchesBegan s'est produit, et ne déplacer que cet objet lorsque touchesMoved se produit.

Implémentez cette fonctionnalité de déplacement.

#### **Survol d'une poubelle**

Lorsqu'un objet est en cours de déplacement et que le doigt passe au-dessus de l'un des containers (il faut les tester tous les trois), on veut passer ce container dans l'état "Highlited". Il suffit pour cela de définir sa propriété isHighlighted à vrai. Au contraire, quand le doigt est en dehors du rectangle du container, il faut passer cette propriété à faux.

Implémentez cette fonctionnalité de survol.

#### **Relâchement des objets (drop)**

L'événement touchesEnded se produit lorsque le doigt quitte la surface de l'écran. Il faut alors vérifier si l'objet qui était en cours de déplacement (s'il y en avait un) se situe au-dessus du container qui lui est associé. Si vous avez respecté l'ordre des objets et des containers à la création des OutletCollections, un calcul simple permet de réaliser ce test.

Si le test est positif, il faut faire disparaître l'objet qui était en cours de déplacement, et incrémenter le nombre d'objets bien triés.

Si le test est négatif, il faut renvoyer l'objet qui était en cours de déplacement à l'endroit où il se trouvait au moment de l'événement touchesBegan.

Implémentez cette fonctionnalité de relâchement des objets.

#### **Message de félicitation final**

Lorsque le nombre d'objets correctement triés atteint le nombre total d'objets (6 dans cette application jouet), on peut afficher un message de félicitation à l'utilisateur. Une solution simple consiste à utiliser la classe UIAlertController. En parcourant sommairement la documentation de cette classe dans la Documentation Développeur de Xcode, mettez en place (trois lignes de code suffisent) une alerte ayant l'aspect suivant. Le clic sur "OK" permet de désactiver le message d'alerte. L'utilisateur peut alors revenir sur l'écran d'accueil.

#### **Définition d'une icône personnalisée à votre application**

S'il vous reste du temps, définissez une icône personnalisée à votre application.

#### **Compression du projet et dépôt sur la plate-forme de cours**

Créez une archive compressée au format .zip de votre projet (le dossier contenant le fichier .xcodeproj et un sous-dossier contenant tous les autres contenus : scripts, images, storyboard, ...)

Déposez ce fichier .zip dans l'[espace de dépôt](https://moodle.univ-lyon2.fr/mod/assign/view.php?id=226717).