

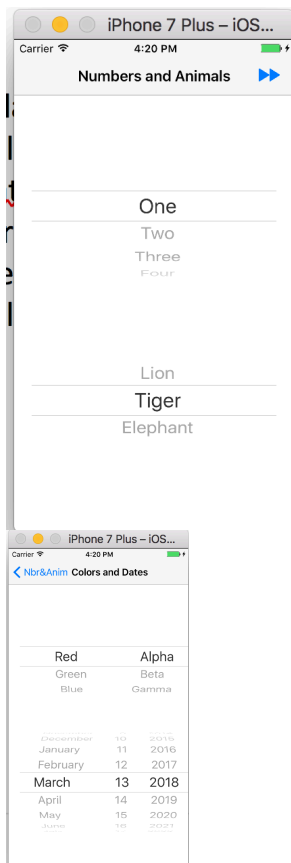
# 23 - UIPickerView StoryBoard

vendredi 9 mars 2018

15:59

## Objectifs du tutoriel

- Créer des listes déroulantes à une ou plusieurs composantes et gérer leur source de données et leur délégué.
- Utiliser un DatePicker (dans le storyboard seulement)
- Intégrer un navigateur contrôleur
- Ajouter des contraintes d'auto-layout
- Internationaliser l'application



1. Créer un nouveau projet SingleViewApp, nommé le "PickerView Storyboard"
2. Configurer ce projet de façon standard;
  - a. Ajouter les icônes du collège
  - b. Restructurer les fichiers dans le groupe supporting file
  - c. Enlever les références au Launchscreen storyboard et supprimer ce fichier
  - d. Supporter toutes les orientations, upside down y compris
  - e. Ajouter les traces NSLog au appDelegate, créer les 2 schémas de compilation debug et release

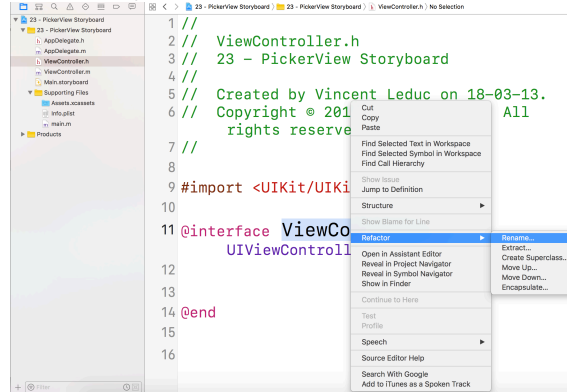
### 3. Le storyboard

Mis à part les délégués des listes déroulante (qui seront créés plus loin dans le tutoriel) et le contrôleur de navigation (pour supporter le upside down), tout est fait dans le storyboard;

#### a. Le contrôleur de navigation

##### i. Renommer le contrôleur de vue créé par défaut "NavigationController"

(n'oubliez pas les commentaires)



##### ii. Modifier son .h pour qu'il dérive de UINavigationController

```
11 @interface NavigationController :
    UINavigationController
```

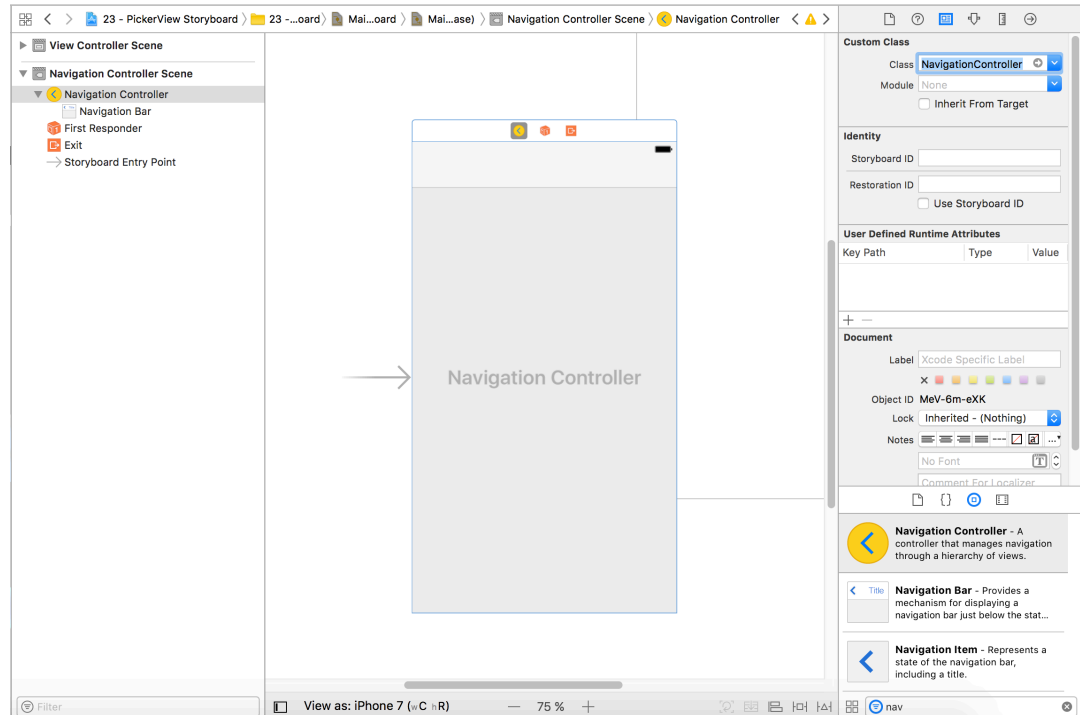
##### iii. Dans son .m ajouter code supportant le upsidedown

```
30 -(UIInterfaceOrientationMask)
    supportedInterfaceOrientations
    {
31     return UIInterfaceOrientationMaskAll;
32 }
33
34 -(BOOL)shouldAutorotate {
35     return YES;
36 }
```

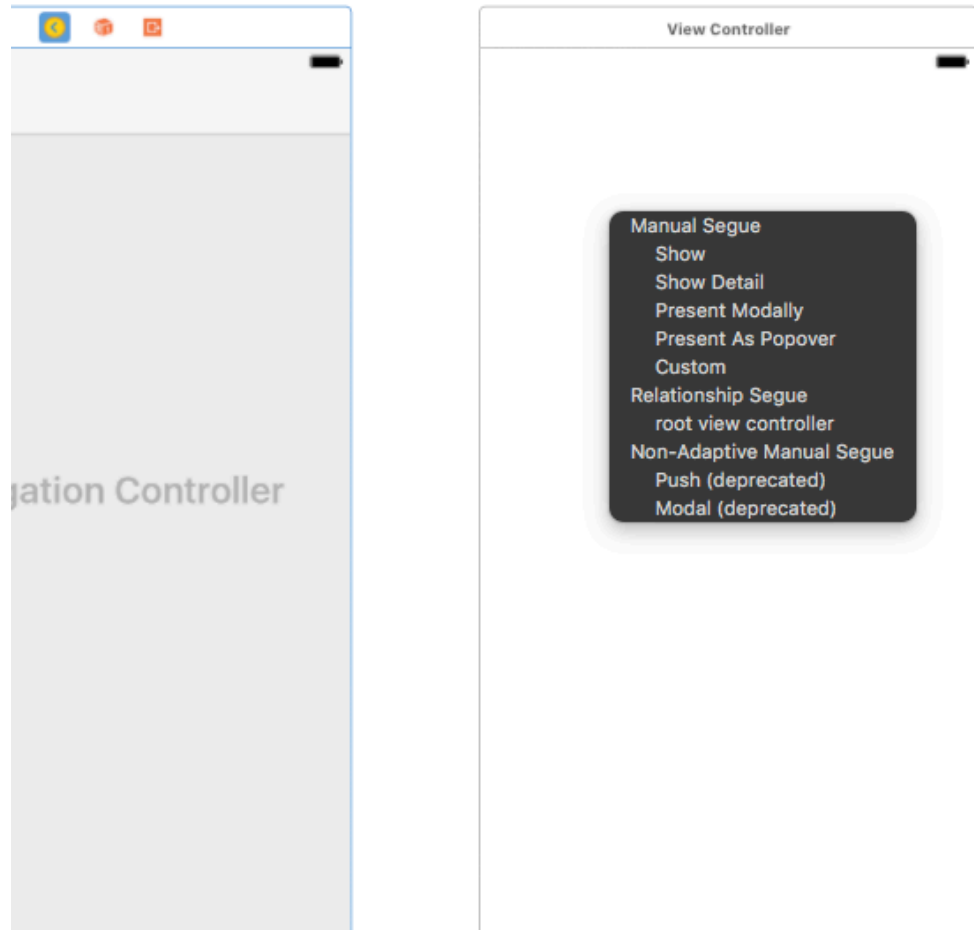
##### iv. Dans le storyboard, ajouter un Navigation Controller, supprimer sa scène root créée par défaut (nous verrons le Table View plus tard)

##### v. Assurer vous que le Navigation contrôleur est le contrôleur initiale

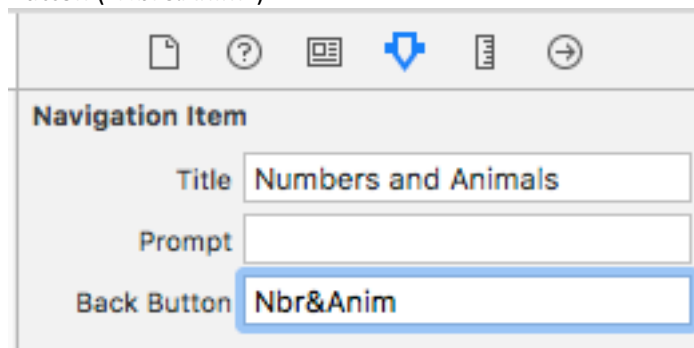
##### vi. Associer cette scène avec la classe NavigationCotroller



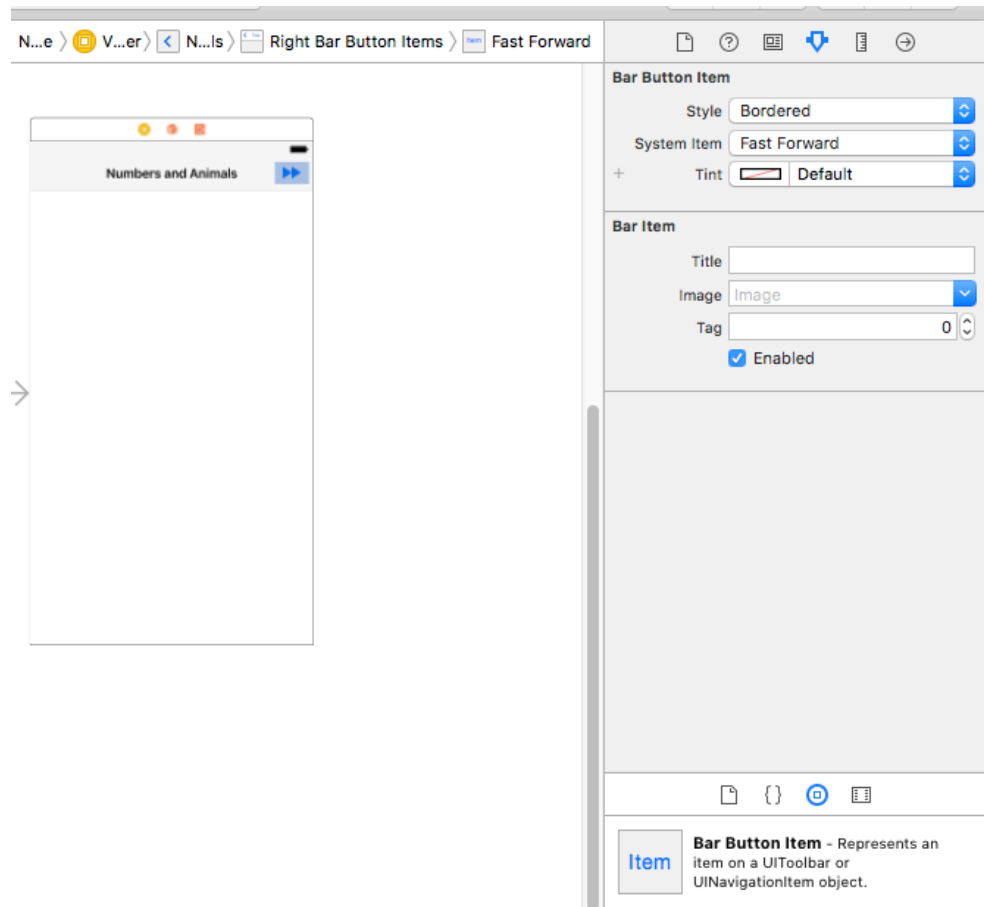
- b. Première Scène - Numbers and Animals
  - i. Utilisez le contrôleur de scène créé par défaut comme root contrôleur du contrôleur de navigation (établissee une relationship segue).



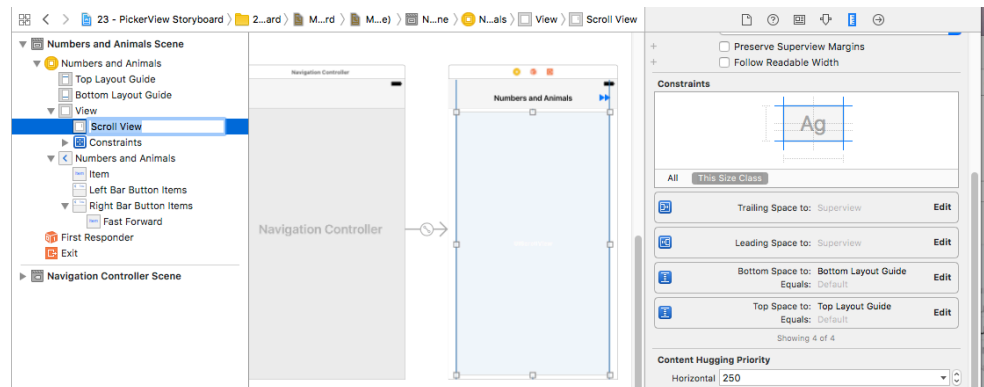
- ii. Donnez un titre au Navigationitem ("Numbers and Animals Scene") et au Back Button ("Nbr&Anim")



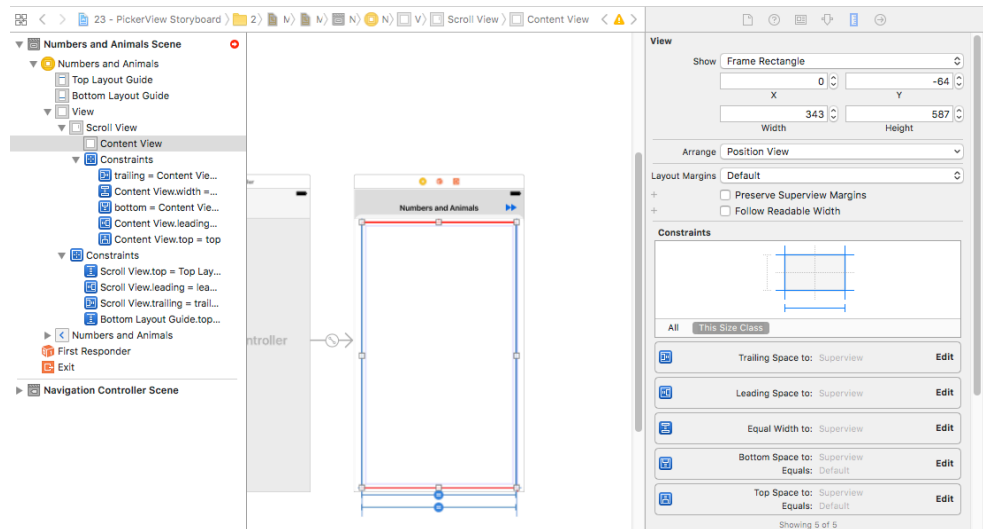
- iii. Ajoutez un Bar Button Item à droite (Fast Forward)



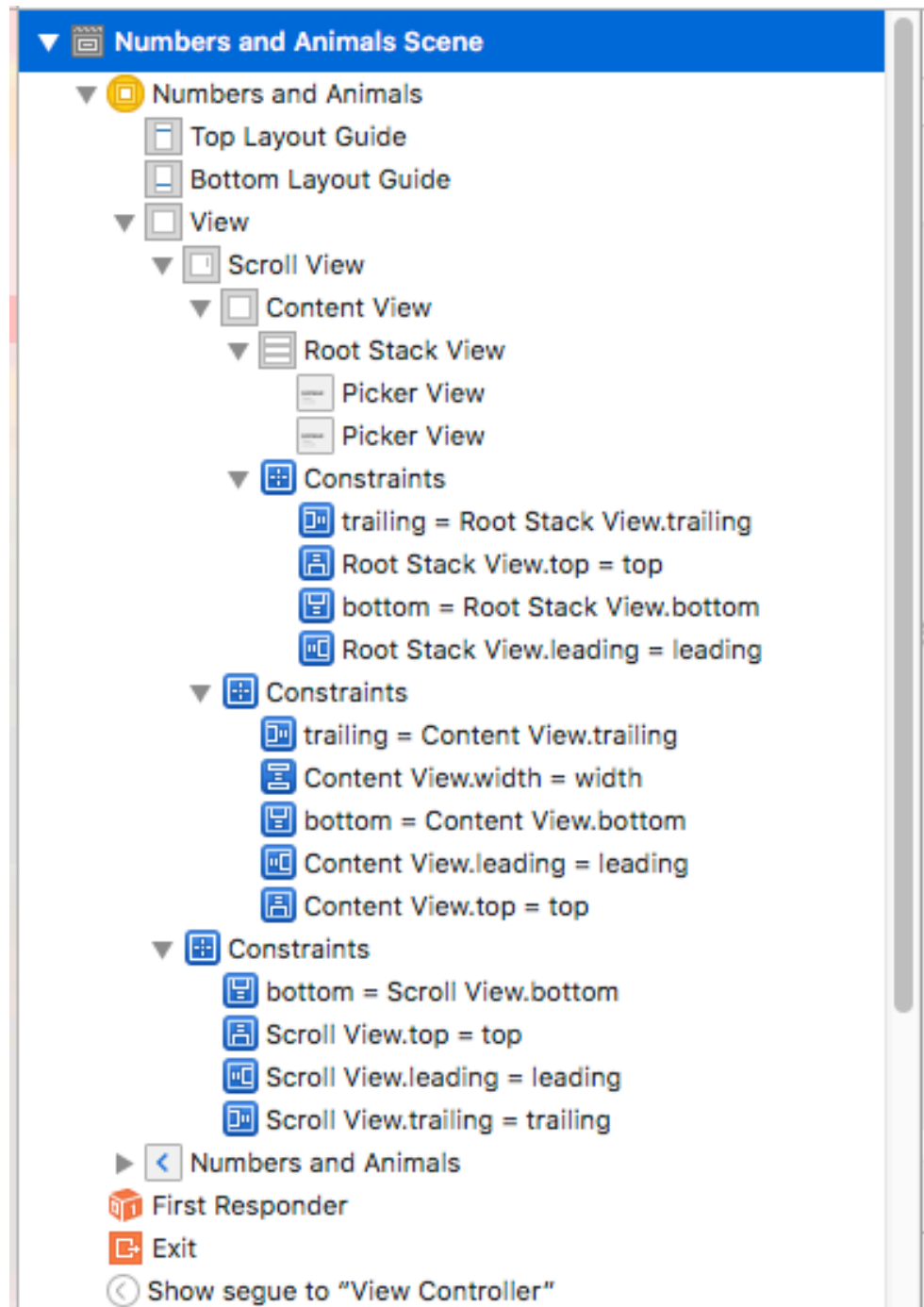
iv. Ajouter une Scroll View accotée à sa super vue



- v. Ajouter une Vue de contenu (Content View)
- vi. Ajuster la position de la Content View en utilisant les guideline de sorte qu'elle s'ajuste sur la Scroll View puis ajoutez 4 contraintes espacement (standard pour top et bottom , 0 pour leading et trailing)



- vii. Désactiver le scroll horizontal en ajoutant une contrainte d'égalité de largeur entre la Content View et celle du Scroll View
- viii. Ajouter 2 Picker View à l'intérieur d'une Stack View Verticale (Root Stack View)
- ix. Accotez votre Root Stack View à sa super View (0 d'espacement bottom trailing et leading, standard pour le top)



- x. Ajuster ses propriétés d'alignement et de distribution de la Root Stack View comme suit

**Stack View**

- + Axis: Vertical
- + Alignment: Center
- + Distribution: Fill Equally
- + Spacing: 8
- + ☐ Baseline Relative

- c. Seconde Scène - Colors and Dates
  - i. Ajouter un second contrôleur de vue connecté par une action segue Show créée à partir du bouton Fast Forward
  - ii. Bâissez son contenu et ses contraintes de la même façon que la scène précédente à la différence près que le second Picker View est un Date Picker View.  
Vous pouvez copier-coller la Scroll View, recréer ses contraintes et remplacer le second UIPickerView par un Date Picker.
4. Les délégués
  - a. NombresPickerViewDelegate
    - i. Ajouter une classe (Cocoa Touch) dérivant de NSObject et implémentant les 2 protocoles UIPickerViewDelegate et UIPickerViewDataSource

Choose options for your new file:

Class: NombresPickerViewDelegate

Subclass of: NSObject

☐ Also create XIB file

Language: Objective-C

Buttons: Cancel, Previous, Next

- ii. Dans le .h, Modifier l'instruction import pour pouvoir utiliser les protocoles du UIKit

```
9 #import <UIKit/UIKit.h>
```



```

11 @interface NombresPickerViewDelegate
    : NSObject <UIPickerViewDelegate,
    UIPickerViewDataSource>

```

- iii. Ajouter une propriété pour le tableau de nombres (non mutable car les données sont statiques dans cet exemple)

```

14 @property (strong, nonatomic) NSArray* nombres;

```

- iv. Dans le.m, sur-définir la méthode d'initialisation de la classe NSObject et initialiser le tableau de nombres avec 4 valeurs

```

15 -(instancetype)init {
16     if(self = [super init]) {
17         [self setNombres:@[@"Un", @"Deux", @"Trois",
18                             @"Quatres"]];
19     }
19     return self;
20 }

```

- v. Implémenter les 3 méthodes obligatoires du protocole de source de données; la méthode appelée à la création de tout UIPickerView qui définit le nombre de composants dans le UIPickerView.

```

22 -(NSInteger)numberOfComponentsInPickerView:
    (UIPickerView*)pickerView {
23     return 1;
24 }

```

la méthode appelée à la création de tout composant dans un UIPickerView pour définir le nombre de lignes dans le composant.

```

26 -(NSInteger)pickerView:(UIPickerView*)pickerView
    numberOfRowsInComponent:(NSInteger)component
    {
27     return [[self nombres] count];
28 }

```

La méthode appelée à la création de chaque ligne du composant pour définir le titre d'une rangée (ici on renvoie simplement l'objet à l'index approprié)

```

30 -(NSString*)pickerView:(UIPickerView*)pickerView
    titleForRow:(NSInteger)row forComponent:
    (NSInteger)component {
31     return [[self nombres] objectAtIndex:row];
32 }

```

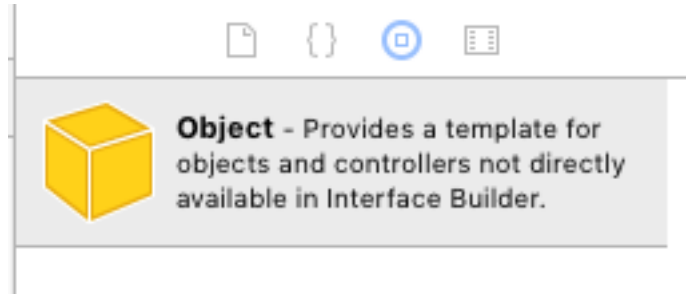
- vi. Finalement implémenter la méthode du protocole UIPickerViewDelegate requise lorsque l'on sélectionne l'un des choix. Ici on se contentera de logger le choix.

```

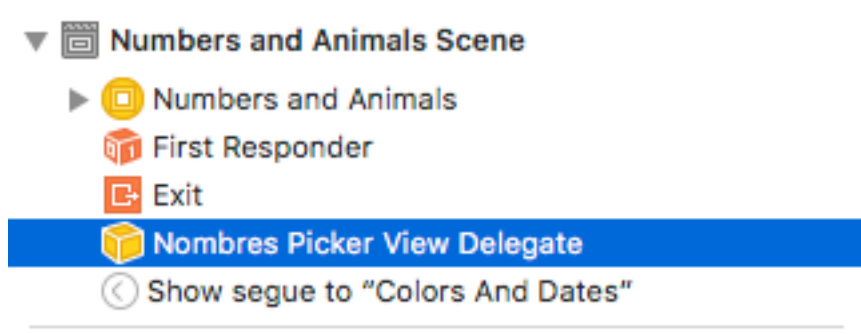
34 -(void)pickerView:(UIPickerView*)pickerView
    didSelectRow:(NSInteger)row inComponent:
    (NSInteger)component {
35     NSLog(@"%@ %@", @"Sélection de l'élément :", [[self
        nombres] objectAtIndex:row]);
36 }

```

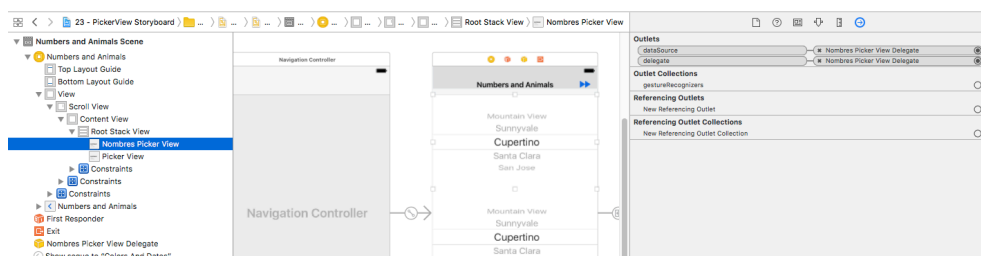
- vii. Connecter la classe d'élève au UIPickerView dans le storyboard  
Ajouter un objet UIKit à la scène contenant le UIPickerView



- ix. Nommer-le Nombres Picker View Delegate



- x. Connecter les 2 protocoles de la Picker View



- xi. Vous pouvez exécuter l'application pour vérifier que la première liste déroulante s'affiche avec des données, se positionne correctement quelque soit l'orientation et que le nombre sélectionnée est bien logué dans la console.
- b. Suivez la même procédure pour créer le délégué AnimauxPickerViewDelegate  
Créer cette fois-ci un tableau de 3 noms d'animaux

```

15 -(instancetype)init {
16     if(self = [super init]) {
17         [self setAnimaux:@[@"Lion", @"Tigre", @"Éléphant"]];
18     }
19     return self;
20 }

```

- c. CouleursPickerViewDelegate; cette liste déroulante dispose de 2 composantes, une pour des couleurs (rouge, vert bleu) et une autre pour 3 types de rayonnement (alpha beta et gamma)
- i. Vous devez déclarer 2 tableaux dans le .h

```

9 #import <UIKit/UIKit.h>
10
11 @interface CouleursPickerViewDelegate :
    NSObject <UIPickerViewDelegate,
    UIPickerViewDataSource>
12
13 @property (strong, nonatomic) NSArray*
    couleurs;
14 @property (strong, nonatomic) NSArray* alphas
    ;
15

```

- ii. Initialiser avec les valeurs suivantes

```

13 @synthesize couleurs;
14 @synthesize alphas;
15
16 -(instancetype)init {
17     if(self = [super init]) {
18         [self setCouleurs:@[@"Rouge", @"Vert", @"Bleu"]];
19         [self setAlphas:@[@"Alpha", @"Beta", @"Gamma"]];
20     }
21     return self;
22 }

```

- iii. Cette fois-ci le UIPickerView définit 2 composantes

```

25 -(NSInteger)numberOfComponentsInPickerView:(UIPickerView*)
    pickerView {
26     return 2;
27 }

```

- iv. Et selon l'indice du composant (0 pour les couleurs, 1 pour les alphas - choix arbitraire) on renvoie la la taille du tableau correspondant

```

29 -(NSInteger)pickerView:(UIPickerView*)pickerView
    numberOfRowsInComponent:(NSInteger)component {
30     return component == 0 ? [[self couleurs] count] :
31         [[self alphas] count];
32 }

```

- v. Le titre de la rangée suit le même principe

```

34 -(NSString*)pickerView:(UIPickerView*)pickerView titleForRow:
    (NSInteger)row forComponent:(NSInteger)component {
35     return component == 0 ? [[self couleurs] objectAtIndex:row] :
36         [[self alphas] objectAtIndex:row];
37 }

```

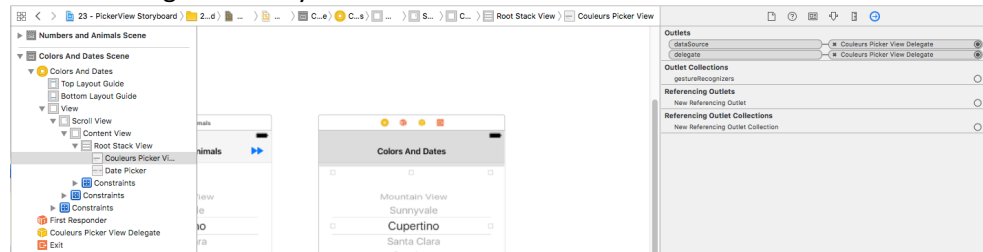
- vi. On utilise la méthode précédente pour logger l'item sélectionné selon la rangée et la composante

```

39 -(void)pickerView:(UIPickerView*)pickerView didSelectRow:
    (NSInteger)row inComponent:(NSInteger)component {
40     NSLog(@"%@ %@", @"Sélection de l'élément :", [self pickerView:
        pickerView titleForRow:row forComponent:component]);
41 }

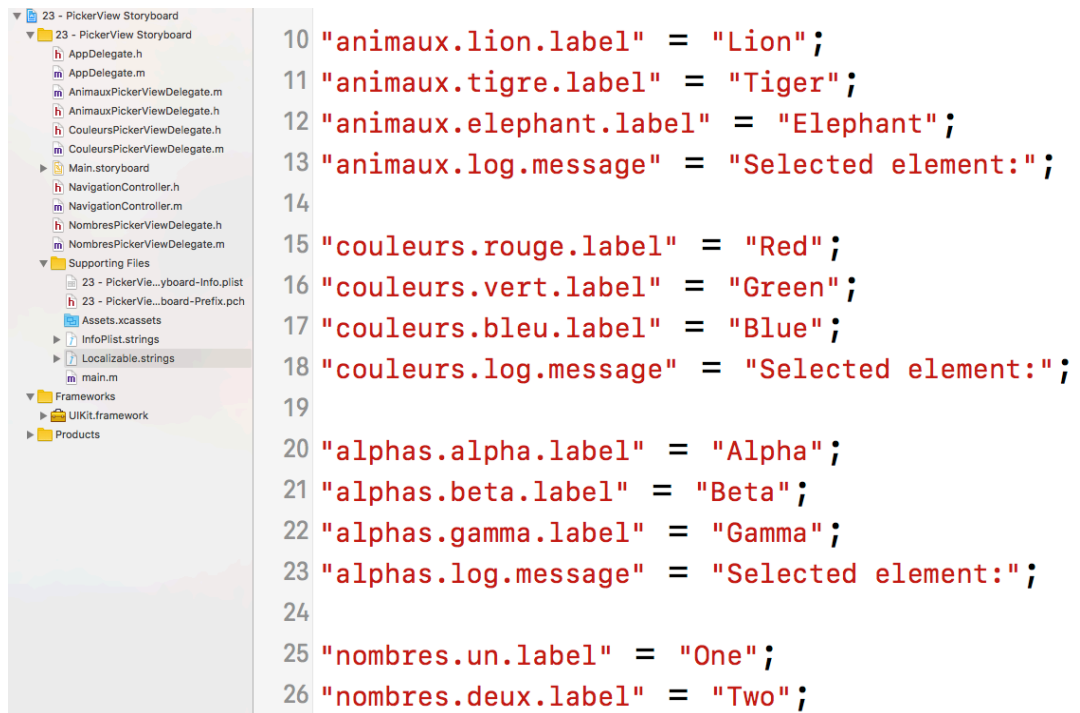
```

- vii. Il reste à utiliser la même technique que précédemment pour connecter le code du délégué au storyboard

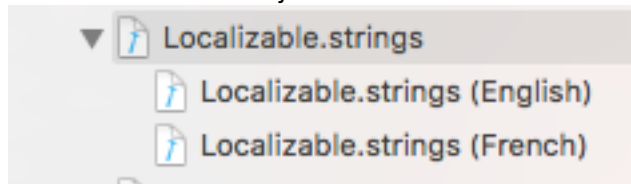


## 5. Internationaliser votre application

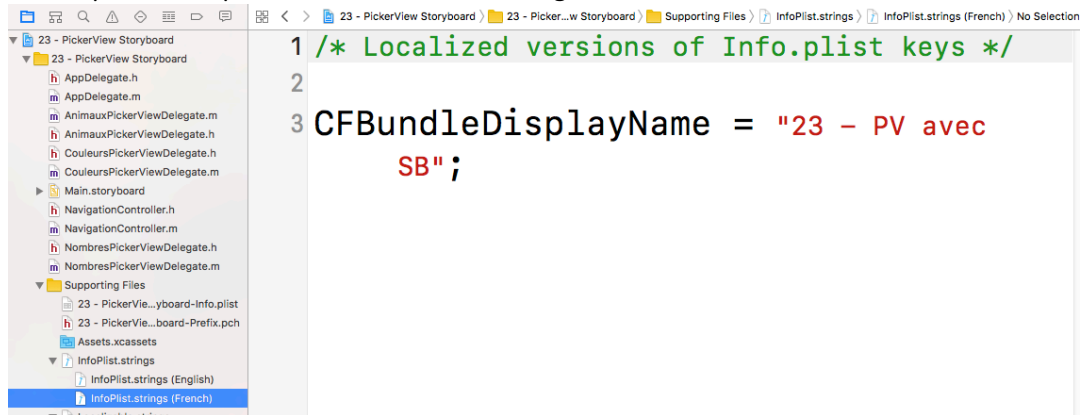
- a. Externaliser vos chaînes de caractères dans un fichier Localizable.strings



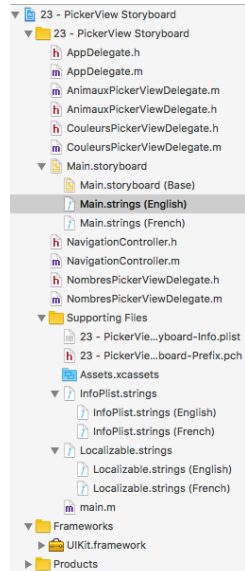
- b. Localiser ce fichier et ajouter vos traductions



- c. Même procédure pour le fichier InfoPlist.strings



- d. Localiser le fichier Main.storyboard



```
1 /* Class = "IBUINavigationController"; title =  
   "Numbers and Animals"; ObjectID = "DTr-  
   Rs-MSB"; */  
2 "DTr-Rs-MSB.title" = "Numbers and Animals";  
3  
4 /* Class = "IBUINavigationController"; title =  
   "Colors and Dates"; ObjectID = "XFv-9Z-  
   J11"; */  
5 "XFv-9Z-J11.title" = "Colors and Dates";  
6  
7 /* Class = "IBUIBarButtonItem"; title =  
   "Nbr&Anim"; ObjectID = "oP1-3F-Qvi"; */  
8 "oP1-3F-Qvi.title" = "Nbr&Anim";  
9  
10 /* Class = "IBUIBarButtonItem"; title =  
   "Col&Dat"; ObjectID = "xL0-z1-9T3"; */  
11 "xL0-z1-9T3.title" = "Col&Dat";
```