
TP 1 : NUMBER

MBDS

31/10/2013

Objectives

L'objectif de ce TP est de vous permettre de réaliser une application simple pour iPhone représentant un petit jeu.

Dans ce jeu, l'utilisateur devra deviner le nombre choisi par l'application, en suivant les indications données.

Un seul écran sera suffisant pour la réalisation de cette application.

Ressources supplémentaires

N/A

Activités

1. Créer et préparer le projet

- a. Créer un nouveau projet de type **Single View Application**
- b. Nommer le **Number**
- c. Ajouter un dossier dans la partie gauche, nommé **src**
- d. Déplacer les 4 fichiers suivant dedans :
 - i. ViewController.m et .h
 - ii. AppDelegate.m et .h

2. Création de l'interface (*Fichier : Main.storyboard*)

- a. Placer un **UILabel** afin de créer le titre de l'application en haut de l'écran : **Number**
- b. Placer en dessous un **UITextField** qui permettra à l'utilisateur de saisir un nombre
 - i. Sélectionner la valeur **Number Pad** pour l'option **Keyboard**, dans la partie droite de votre interface
- c. Placer 2 **UIButton** en dessous :
 - i. Un pour vérifier le nombre entré, nommé **Check**
 - ii. L'autre pour réinitialiser le jeu, et générer un nouveau nombre à deviner, nommé **Reset**
- d. Enfin, ajouter un **UITextView** pour afficher les aides de jeu.

3. Rendre accessible les éléments de l'interface

- a. Partager l'affichage (*icône "Costume" en haut à droite*) afin d'avoir sur la partie gauche le **Storyboard** et sur la partie droite **ViewController.h**
 - i. Effectuer un **ctrl+clic** sur l'**UITextField** et déplacer le trait vers **ViewController.h** en dessous du **@interface**.
 - ii. Relâcher le clic et nommer votre nouvelle variable **numberTF**
 - iii. Faire de même pour l'**UITextView** et nommer la **infoTV**
- b. Dans **ViewController.m**
 - i. Ajouter un **@synthesize** pour vos 2 variables en dessous du **@implementation**
 - ii. Effectuer un **ctrl+clic** sur votre **UIButton Check** vers **ViewController.m** en dessous de la méthode **viewDidLoad**
 - iii. Relâcher le clic et nommer votre nouvelle méthode (*équivalent de onClick*) **checkBtn**
 - iv. Faire de même pour l'**UIButton Reset** et nommer cette méthode **resetBtn**

4. Génération du nombre et affichage du message d'accueil

- a. Ajouter dans le fichier **ViewController.h** entre { } en dessous de **@interface**
 - i. **int number;**
- b. Dans le fichier **ViewController.m**, dans la méthode **viewDidLoad**
 - i. Générer un nombre aléatoire compris entre 0 et 99 :
 1. **number = arc4random() % 100;**
 - ii. Afficher un message invitant l'utilisateur à jouer en utilisant **setText** sur **infoTV**

5. Gestion du bouton Reset

- a. Dans la méthode **resetBtn** générer un nouveau nombre aléatoire et effacer le contenu d'**infoTV** en le remplaçant par le message de début de partie.
- b. Il y a t'il un moyen de simplifier ce code ?

6. Gestion du bouton Check

- a. Commencer par masquer le clavier au début de la méthode **checkBtn** :
 - i. **[numberTF resignFirstResponder];**
- b. Récupérer la valeur entrée par l'utilisateur dans **numberTF** et la "**caster**" en **int** (*intValue*)
- c. Effectuer les tests nécessaires pour savoir si le nombre est plus petit, égal ou plus grand que la variable **number**
- d. Pour chaque cas, récupérer le contenu de la variable **infoTV**, concaténer le message correspondant et afficher la nouvelle **NSString** dans **infoTV**.
- e. Il y a t'il un moyen de simplifier ce dernier code ?

7. Faire en sorte d'afficher le nombre d'essai quand l'utilisateur trouve la bonne réponse.