

LABORATOIRE 2 : LE JEU DU PENDU

OBJECTIF

Le but de ce laboratoire est de mettre en pratique les notions au cours iOS sur les appels réseaux.

ÉVALUATION

Le laboratoire est noté selon la qualité du code et du projet avec le barème ci-dessous. Vous devez remettre un fichier zip qui contient votre projet et le nommer :








NNNp_NNNp_201234567_201234567_labo2.zip,

où 'NNN' sont les trois premières lettres de votre nom de famille, 'p' est la première lettre de votre prénom, et 201234567 est votre matricule.

RESSOURCES

- 1) Persistance des données : [lien](#)
- 2) Les listes de données en iOS : [lien](#)
- 3) Appels réseaux : [lien](#)
- 4) App réactive : [lien](#)
- 5) Sauver et charger un dictionnaire dans « UserDefaults » :
 - a. Basic : [HS lien](#)
 - b. Avancé : [lien Cocoacasts](#)

TRAVAIL DEMANDÉ

Partie en 10 coups	Stéphane	Programme
		
1	« T »	--T-----
2	« S »	
3	« O »	-OT-----
4	« G »	
5	« B »	
6	« M »	
7	« R »	-OT-RO-
8	« L »	
9	« V »	
Stéphane a perdu -- le mot était potiron		

Vous devez coder un jeu du pendu : [lien](#). Ce jeu consiste à deviner un mot lettre par lettre en faisant moins de six erreurs en devinant des mauvaises lettres.

Important : Vous n'êtes pas obligé de dessiner le pendu dans votre app, vous pouvez simplement afficher un compteur d'erreurs (0/6, 1/6, etc.).

Votre app aura deux modes de jeu (le fonctionnement est décrit plus bas):

1. un jeu de pendu avec des mots de dictionnaire en anglais,
2. un jeu avec des titres de films

On accède aux modes de jeu à travers des onglets d'un « tabController ». Le « tabController » contient trois onglets : les deux modes de jeu, et un onglet « stats » (pour les statistiques)

Voici les deux API à utiliser et le fonctionnement du jeu associé:

1. Mot de dictionnaire : <https://random-word-api.herokuapp.com/word>
 - Fonctionnement classique
2. Titre de film : <https://www.omdbapi.com/>
 - Fonctionnement classique mais il y aura des indices qui s'affiche en plus
 - Utiliser la recherche par id avec une valeur aléatoire entre tt0000001 et tt0055252
 - Après deux erreurs, vous devez donner l'année de parution (champ « Released »)
 - après quatre erreurs, vous devez donner le « rating », et le « genre »
 - après 5 erreurs, vous devez, idéalement, donner le réalisateur (maximum deux) et les acteurs (maximum trois)

Sauvegarde des données :

Si un utilisateur joue pour la première fois ou bien s'il bat un meilleur score alors une « modal » apparaît et alors le joueur peut entrer son nom. Son nom, son type de jeu (mot de dictionnaire ou titre de film), et son score sont alors sauvegarder de manière persistante.

Vous pouvez utiliser UserDefaults(), ou un fichier. L'utilisation de CoreData est réservée pour la section « Avancée ».

À vous de choisir le modèle de données qui convient.

Avancée :

Choisissez une fonctionnalité à implanter parmi les deux ci-dessous :

1. Utiliser CoreData pour sauver les historiques de partie, et afficher les 5 meilleurs scores dans l'onglet « stats ». Vous pouvez utiliser un « segmented controller » pour afficher les meilleurs scores selon les types de jeu.
2. Ajouter une animation personnalisée lorsque la modal pour entrer le nom du joueur apparaît ([lien](#))
3. Ajouter une suite de test unitaires pour vos modes de jeu (dans le « Model »)

BARÈME

Important : Vous n'êtes pas obligé de dessiner le pendu dans votre app, vous pouvez simplement afficher un compteur d'erreurs (0/6, 1/6, etc.).

Voici le barème approximatif :

	Barème [%]
Mode de jeu « Mot du dictionnaire »	60
Mode de jeu « Titre de films »	10
Une fonctionnalité avancée	5
Respect du modèle MVC	10
Qualités du code et de la app	15