


Projet 3 de la formation iOS

Objet :
Présentation du Projet 3





Ce document est réalisé pour expliquer l'architecture de mon projet 3 réalisé dans le cadre de ma formation iOS Openclassrooms. Vous y trouverez l'architecture de mon projet et le rôle de chacune des classes.

1. main.swift

Ce fichier est la colonne vertébrale du projet, c'est par lui que nous allons faire appel aux différentes fonctions et classes dont nous avons besoin pour faire fonctionner le projet. Tout débute par une boucle qui permet aux joueurs de sélectionner leur trois combattants dans une liste ainsi que leur nom. Une vérification à lieu afin de s'assurer que le choix du joueur est correct et si le nom n'existe pas déjà. Une fois cette boucle finie, la classe Fighter est appelée afin d'initialiser les personnages puis c'est autour des joueurs au travers de la class Player.

Une fois cette étape réalisée, on rentre dans une boucle surveillant que la partie n'est pas terminée, par le biais d'un boolean. Cette boucle permet de gérer les actions du jeu, à savoir faire combattre deux joueurs ou en soigner un. Les fonctions qui assurent ces actions sont dans la classe Game.

2. message.swift

Ce fichier regroupe les messages qui sont récurrent au cours de la partie. La mise en place d'un switch case permet de regrouper ces messages et de les faire apparaître à tout moment du jeu sans avoir à les affilier à des classes ou de les réécrire en dur dans le code.

3. function.swift

Cette partie, appelée fonction, correspond à un ensemble de fonction qui permet de s'assurer que le jeu se déroule dans les bonnes conditions et n'appartiennent à aucune classe, c'est pourquoi elles sont regroupées dans un dossier à part.

4. Player.swift

Cette classe permet d'initialiser un joueur avec le nom de son équipe ainsi que les membres qui la compose. Une fois cette classe initialisée nous allons pouvoir réaliser les opérations de combats entre les différents fighter par le biais de cette classe.

5. Fighter.swift

Cette classe permet d'initialiser les informations propre d'un combattant, comme son nom, ses points de vies etc. Cette classe est appelée lorsque le joueur sélectionne au début de sa partie un combattant.

6. Game.swift

La classe Game permet de gérer l'ensemble des paramètres du jeu, à savoir le déroulement de la partie mais également l'affichage dans le terminal.

—> Le déroulement de la partie :

- choiceAttackOrCare : Cette fonction permet aux joueurs l'action à appliquer à chacun de leur tour, à savoir attaquer ou se soigner.
- recordGame : Cette fonction regroupe les informations tour par tour afin de faire un résumé aux joueurs une fois que la partie est finie.
- chest : cette fonction permet de générer aléatoirement un coffre qui peut contenir trois objets différents, obtenues de manière aléatoire.
- turnMore : Cette fonction permet d'incrémenter tour par tour une variable turn.
- endGame : Cette fonction renvoie un booléen, c'est elle qui permet de savoir si la partie est terminée ou non.

—> Affichage dans le terminal :

- clearConsol : permet de faire disparaître les informations du tour précédent
- startGame : initialise le début de la partie avec un message d'accueil
- fighterMenu : permet d'afficher les choix de personnages disponibles.
- printScenario, spaceInterface, messageBuilder et printMenu : permettent d'afficher le tableau résumant l'état des membres des deux équipes.