****

**LO52**

**RAPPORT DE PROJET**

**-**

**APPLI F1 RACE**

***Auteurs :***  ***Encadrants :***

GAUJOUR Bastien BRISSET Fabien PALERMO Enzo ROMET Pierre

**SOMMAIRE :**

**Cahier des charges :**

Classement/Statistiques :

* Meilleur temps par coureur :
  + Au Pitstop 1
  + Au Fractionné
  + Au Sprint
  + Au Cycle
* Meilleur temps par équipe :
  + Au Pitstop 1
  + Au Fractionné
  + Au Sprint
  + Au Cycle
  + En temps total (Somme des cycles ou des ateliers)

Bonus : Fonction historique des courses précédentes (nécessite la création d’une course par l’user avant l’équipe)

Chronométrage :

* Manuel :
  + 1 Appui pour le départ d’un coureur
  + 1 appui pour le split de chacun de ses ateliers.

Il y a donc un chrono manuel par équipe.

* Auto (Bonus):
  + Chrono général qui démarre au départ du coureur 1 et qui s’arrête au dernier split du dernier coureur. Indique le temps total de la course

**Attention :** Les équipes peuvent concourir en même temps, il peut donc y avoir x chrono manuels simultanés si x équipes.  
(Pas de gestion du multitouch cependant donc les départs resteront différés)

Setup de la course :

* Rentrer manuellement chaque joueur, avec nom, prénom, niveau (de 1 à 100).  
  Le nombre total de participant devra être un multiple de 3 (indiquer le nombre total en direct à l’user)
* Générer des équipes équilibrées de 3 coureurs, de telle façon que l’écart entre l’équipe la plus forte et la plus faible ne dépasse pas 25.
* Rentrer manuellement l’ordre de passage des équipes, et des coureurs dans chaque équipe

**Application :**

Interface de l’application :  
Création des participants :   
On pourrait lui indiquer en direct combien de participant il manque selon le total actuel ??

L'application se décompose en plusieurs fragments correspondants chacun à un écran.   
Dans l'ordre cela donne :

**Présentation :**  
- Bouton création course

**Création course :**  
- date : préremplir les infos au moment de laffichage  
- heure : préremplir les infos au moment de laffichage   
- nom : non obligatoire ? (Identification par la date sinon)

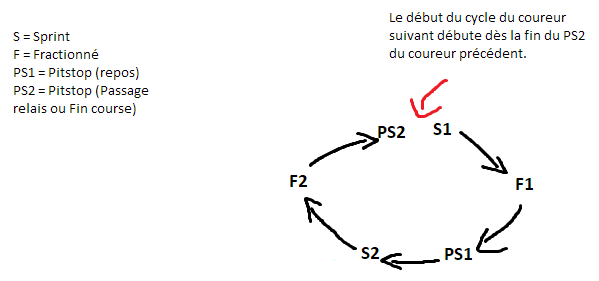
**Création des participants :**  
- Nom  
- prénom  
- niveau  
*(Il y aura une petite note explicative pour que l’user rentre un multiple de 3)*

(Génération des équipes, si trop long ajouter un petit écran de chargement)

**Gestion des ordres de passage :**  
- Ordre de passage des équipes  
- Ordre de passage des coureurs dans une équipe

**Gestion de la course :**  
- Onglet Gestion course :  
 - Chrono général de la course : du premier départ à la dernière arrivée  
 *(Ajouter une note : le chrono débutera dès le premier départ)*  
 - Team boxes : Num d'équipe, nom coureur, atelier en cours, chrono de l'équipe  
 *(A chaque clic sur le bouton split, on récupère le temps de l'atelier)*

- Onglet Classement :  
 - Accès au classement par team en direct (Par temps total)  
 - Accès au classement par coureur en direct (Par temps de cycle) (+ possibilité de voir le classement par atelier ?)  
 - Accès aux infos d'une équipe/coureurs lors du clic :  
 - Clic équipe (depuis classement équipe) : coureurs qui la composent + leurs temps par cycle.  
 - Clic coureur (depuis classement coureur) : leurs temps par atelier



Algorithme de création des équipes :  
  
Il est (probablement ?) impossible de trouver une formule qui satisfait la condition à coup sur quel que soit le tirage, puisque la construction de formule se fait par rapport à un tirage.  
Nous avons donc essayé de modifier les valeurs après une première tentative de création la plus équilibrée possible.  
  
L'algorithme peut se décomposer en plusieurs étapes :

*Etape 1 : Création des premières équipes :*  
  
Sois une liste de valeurs des niveaux des joueurs, ordonnée par ordre croissant :  
16,20,33,35,36,56,57,65,66,69,75,80,81,90,95,98,99,100  
  
On forme les équipes de la façon suivante :  
 - On prend le plus petit élément, le plus grand élément et l'élément médian (En cas de nombre pair, élément médian -1)  
 - On forme l'équipe n°1 = total de ses niveaux et on les retire de la liste (Ex ici équipe n°1 = 16 + 100 + 66 = 182)  
 - …etc jusqu'à vider la liste

*Etape 2 : Premier test de l'écart :*Ensuite, on créé une liste avec les valeurs totales de chaque équipe, puis on l'ordonne par ordre croissant, et on redéfini le numéro de l'équipe selon la position dans cette liste.  
Dans notre exemple cela donne 182, 183, 188, 196, 205, 217  
  
On réalise un test de l'écart entre la plus forte et la plus faible, ici il est de 35 et dépasse donc la valeur seuil de 25.

*Etape 3 : Ajustement n°1 :*  
  
On prend donc l'équipe médiane +1 (ici 196, équipe n°4) et l'équipe la plus forte (ici 217, équipe n°6), puis on échange leurs valeurs médiane :  
Avant échange :   
 E4 = 33 + 98 + 65 = 196  
 E6 = 56 + 81 + 80 = 217  
Après échange :  
 E4 = 33 + 98 + **80 = 211**  
 E6 = 56 + 81 + **65 = 202**

On ré-ordonne les équipes et redéfini leurs positions :  
182, 183, 188, 202, 205, 211  
Puis on reteste l'écart : il est de 29.  
  
*Etape 3 : Ajustement n°2 :*

On continue donc l'échange de valeur médiane, mais cette fois-ci entre la valeur médiane -1 (188, E3) et la plus forte (211, E6).  
(Il a fallu utiliser médiane -1 et non médiane +1 car les valeurs 202 et 211 correspondaient déjà aux deux équipes récemment créées, il y aurait donc eu une étape inutile qui aurait annulé l'échange précédent)  
Avant échange :  
 E3 = 20 + 99 + 69 = 188  
 E6 = 33 + 98 + 80 = 211  
Après échange :  
 E3 = 20 + 99 + **80** **= 199**  
 E6 = 33 + 98 + **69 = 200**Nous avons donc le nouveau classement 182, 183, 199, 200, 202, 205  
Ce qui nous donne un écart de 23 < 25.

On réalise autant d'ajustement que nécessaire jusqu'à réduction de l'écart sous le niveau des 25.

Bases de données :  
  
Afin de stocker de façon permanente les données de l'application, il est nécessaire d'utiliser des bases de données.  
  
Cette fonctionnalité nous sera notamment utile pour consulter les différents classements, mais aussi pour consulter l'historique des précédentes courses.  
Cela permet également de clarifier la gestion des données de l'application dans le programme.

Nous utiliserons la fonctionnalité **Room** d'android studio qui permet la gestion de bases de données facilement en intéragissant avec des objets.

( https://developer.android.com/codelabs/kotlin-android-training-room-database#0)  
Il y aura donc 3 bases de données différentes :

*Race db :*

Cette db contiendra les champs suivants : ID | RaceName | DateCreation | HeureCreation | Teams db | Runners db   
  
Les champs "Teams db" et "Runners db" serviront à stocker les 2 databases suivantes afin de consulter l'historique des courses.

La db sera initié sur l'écran de création de course.

*Runners db :*

Cette db contiendra les champs suivants : ID | Nom | Prenom | Niveau | Order | tps 1 | tps 2 | tps 3 | tps 4 | tps 5 | tps cycle  
  
Les champs "tps 1…5" et "tps cycle" serviront aux fonctionnalités de classements et d'historique. Ils sont initiés à 0.  
Le champ "order" défini l'ordre de passage du joueur dans son équipe, de 1 à 3

La db sera initié sur l'écran de création des participants.

*Teams db :*

Cette db contiendra les champs suivants : ID | Order | ID Runner 1 | LevelRunner1 | ID Runner 2 | LevelRunner2 | ID Runner 3 | LevelRunner3 | Niveau total | tps total  
  
Le champ "tps total" sert lui aussi aux fonctionnalités de classement et d'historique. Il est initié à 0.  
Le champ "order" défini l'ordre de passage de l'équipe, de 1 au nombre d'équipes.

La db sera initié après l'algorithme de création d'équipe, entre l'écran de création d'équipe et de gestion de l'ordre de passage.

*Note :* le chronomètre de chaque équipe est un décompte en secondes, qui sera converti en string formattée MM:SS:ms pour être stocké dans la db et affiché à l'user.  
Cela permettra de simplifier les calculs d'écart de temps et résout le problème de type pour le stockage du temps dans la base de données (Pas de format Time pris en charge par Room)  
  
*Note 2 :* Il a fallu utiliser le plugin "kapt" pour que les databases fonctionnenent avec kotlin. Il y avait un crash de l'application à la création.  
(Ajouter apply plugin: 'kotlin-kapt' dans le fichier build.gradle et remplacer "androidprocessor" par "kapt")