

# PF: Rattrapage 2019

## Sujet: Réalisation d'un jeu de sudoku et de son solver

Vous trouverez dans ce document votre sujet de rattrapage d'Introduction à la programmation fonctionnelle. Ce rattrapage doit vous permettre de démontrer vos compétences en terme de développement de programmation fonctionnelle. Vous trouverez ci-dessous le sujet de votre rattrapage.

### Contexte

Le sudoku est un jeu devenu populaire ces dernières années. Ce jeu "est un [jeu](#) en forme de grille défini en [1979](#) par l'[Américain Howard Garns](#), mais inspiré du [carré latin](#), ainsi que du [problème des 36 officiers](#) du [mathématicien suisse Leonhard Euler](#).

Le but du jeu est de remplir la grille avec une série de chiffres (ou de lettres ou de symboles) tous différents, qui ne se trouvent jamais plus d'une fois sur une même ligne, dans une même colonne ou dans une même région (également appelée « bloc », « groupe », « secteur » ou « sous-grille »). La plupart du temps, les symboles sont des chiffres allant de 1 à 9, les régions étant alors des carrés de  $3 \times 3$ . Quelques symboles sont déjà disposés dans la grille, ce qui autorise une résolution progressive du problème complet." Wikipedia.

### Travail à réaliser

Vous devez réaliser un jeu de sudoku en Scala **en respectant les principes de la programmation fonctionnelle**. Concrètement, votre jeu doit être ergonomique et utilisable en ligne mode console. Vous devez créer un projet complet **avec une approche TDD**. En d'autres termes, vous devez écrire les tests que doivent satisfaire votre application avant d'en écrire la logique.

### Etape 1: Mécanisme de jeu

Le sudoku se joue à un joueur, il vous faudra afficher la grille à chaque étape et permettre au joueur de faire des propositions et des validations:

- ☐ On entend par proposition le fait qu'un joueur puisse donner un ou plusieurs chiffres pour une case comme étant des valeurs candidates (c'est à dire qui peuvent d'avérer correcte pour cette case)
- ☐ On entend par validations, un seul et unique chiffre qui est la valeur déduite par le joueur.

A titre d'analogie, certains joueurs de sudoku notent au crayon à papier les valeurs possible pour une case puis ils note au stylo la valeur définitive.

Il vous faut donc définir:

- Des grilles correctes à résoudre
- Les mécanismes de jeu nécessaire pour permettre à un joueur de résoudre le sudoku

## Etape 2: résolution

Vous devez adjoindre à votre application un "solver", c'est à dire une fonctionnalité permettant de résoudre la grille de sudoku.

Vous devez fournir trois fonctionnalités à un utilisateur

1. La possibilité de demander vérification de la valeur d'une case précise  
Le programme doit dire si la valeur X est correcte pour la case
2. La possibilité de donner la valeur pour une case précise  
Le programme retourne la bonne valeur pour la case et affiche la grille mise à jour
3. La possibilité de donner à l'utilisateur la résolution complète de la grille  
Le programme affiche la grille complète avec les bonnes valeurs. Les validations bonnes de l'utilisateur sont en vert, les mauvaises validations de l'utilisateur sont en rouge, le reste reste en blanc

## Contraintes techniques:

- ☐ Le projet doit respecter les principes de la programmation fonctionnelle
- ☐ Le projet doit être réalisé en Scala 2.12.8
- ☐ Le projet doit utiliser SBT
- ☐ Le projet doit être réalisé en TDD, une part non négligeable de l'évaluation se fera sur la qualité des tests

## Plannification et Livrables

### Livrables:

- Un rapport synthétique expliquant votre modélisation (avec schema), la manière donc fonctionne votre application et une explication très détaillées de comment fonctionne votre solver
- L'url de votre dépôt Git contenant le code source (si dépôt github privé, merci d'ajouter aca-polytech comme utilisateur)

### Date:

- Le projet complet doit être livré d'ici le **dimanche 19 mai 2019 23h42**