

# Programmation Fonctionnelle Avancée

## Cours 10

kn@lri.fr

<http://www.lri.fr/~kn>

# Étude de cas



Afin de consolider nos connaissances, nous allons résoudre un problème qui mobilise tout ce qu'on a vu depuis le début de l'année.

On va essayer de trouver ensemble l'algorithme général. Il faudra ensuite le recoder en TP + les fonctions auxiliaires non présentées.

# Boggle™



Le Boggle est un jeu de mots se présentant sous la forme d'une boîte contenant 16 dés portant des lettres:



À chaque tour, les lettres sont mélangées puis les joueurs ont une minute pour trouver le plus de mots possibles. Les règles sont les suivantes :

- ◆ On peut former un mot en utilisant des cases adjacentes (côté ou diagonale)
- ◆ Une même case ne peut être utilisée deux fois dans un même mot

Ici : SONGE (ok), SONO (pas ok, on ne peut pas utiliser 2 fois le O)



Problème à résoudre :

*Écrire un programme qui génère une grille aléatoire de taille  $N \times N$ , puis affiche la grille, puis tous les mots valides sur cette grille avec les règles du Boggle.*

On dispose d'un fichier `french_simple.txt` des mots français sans accents

# Réflexion collective



Vous pouvez faire appel à tout ce qu'on a vu depuis le début de l'année

Proposer une façon des façons d'aborder le problème (il y a plusieurs choses à considérer).

Votre solution doit être compétitive (une grille de taille  $N = 100$  doit être résolue en moins d'une seconde)

# Go!

# Comment procéder ?



- ◆ Trouver un type de données pour l'entrée
- ◆ Essayer sur un exemple simple (grille 3x3) une solution naive
- ◆ Tenter d'estimer le coût de cette solution sur le problème de taille finale. Si c'est ok, pas besoin de chercher plus loin
- ◆ Sinon, voir ce qui est coûteux et si où on peut gagner en temps de calcul/mémoire.

