Projet 10: Explication du bonus

```swift
func tableView(_ tableView: UITableView, willDisplay cell: UITableViewCell, forRowAt indexPath: IndexPath) {
    if isFirstLoad[indexPath.item] != false {
        cell.transform = CGAffineTransform(scaleX: 0.1, y: 0.1)
        UIView.animate(withDuration: 0.7, delay: 0.0, usingSpringWithDamping: 0.5, initialSpringVelocity: 0.5, options:
            [.allowUserInteraction], animations: {
            cell.transform = .identity
        }, completion: nil)
    }
    isFirstLoad.updateValue(false, forKey: indexPath.item)
}
```

J'ai choisi de créer une animation dans les cellules de la tableView qui affiche les éléments de la recette (photo, ingrédients, temps de cuisson, note) à l'apparition des cellules et une seule fois par recette affichée.

L'animation se jouera à l'apparition des cellules, c'est à dire lorsque l'utilisateur fera l'action de *scroller* la tableView pour voir les différentes recettes.

La variable isFirstLoad est un dictionnaire [Int: Bool] qui a pour clé le numéro de l'item et pour valeur un Booléan, si la celle a déjà été affichée, la valeur est mise à jour avec False.

On applique un CGAffineTransform sur la cellule que l'on met à l'échelle 0.1, 0.1.

On gère ensuite l'animation avec une durée, un damping et une option .allowUserInteraction car on souhaite que l'utilisateur puisse sélectionner une recette (action de tapper sur une cellule) sans que l'animation soit terminée (sinon on perd en UI).

Lorsque l'animation est terminée, comme indiqué précédemment on met à jour la valeur du dictionnaire pour la clé correspondante afin de ne pas rejouer l'animation si l'utilisateur repasse sur la même recette en *scrollant*.

L'objectif de ce bonus est de créer une animation agréable qui permet de rendre l'expérience utilisateur encore meilleure lorsque celui-ci *scroll* pour découvrir de nouvelles recettes.