

Fiche 4.0 : récursivité

BUT Informatique
IUT de Vélizy

1 Empilements

- 0. **Prise en main - empilements0.** Lisez le code et exécutez-le. Cet exercice utilise une mini-bibliothèque `libEmpilements` qui étend `tkiteasy`; les exercices qui suivent seront faits sur le même modèle. L'objectif de cette question est seulement de comprendre l'utilisation de la méthode `empilerCube`. Modifiez librement le script en ajoutant des cubes supplémentaires empilés avec les premiers ou sur des espaces vides.
- 1. **Trois fonctions - empilements1.** Ce script définit trois fonctions qui s'appellent entre elles. L'objectif ici est de comprendre en regardant le code pourquoi les cubes tombent exactement dans cet ordre et sur ces positions.
- 2. **Sans tricher! - empilements2.** Si vous avez compris, essayez de prévoir dans quel ordre et à quels endroits vont tomber les cubes si l'ont fait l'appel de fonction `fa(g,2)` (il faut l'ajouter). Prenez un papier et essayez de noter dans quel ordre tombent les cubes, sur quel emplacement et avec quelle couleur! Si vous vous êtes trompé, ce n'est pas grave mais essayez de comprendre pourquoi!
- 3. **Cette fois-ci, vraiment sans tricher! - empilements3.** Même exercice si l'on ajoute dans ce script l'appel de fonction `fA(g,2)`.
- 4. **Chassé-croisé - empilements4.** Essayez de prévoir ce que va faire l'appel `f0(g,0)`.
- 5. **suite - empilements4.** A la fin de la fonction `f0`, ajoutez l'empilement d'un cube rouge sur la même position que le cube bleu. Comment et dans quel ordre cela va-t'il se passer?



Dans l'exercice précédent, vous avez vu deux fonctions `f0` et `f1` qui s'appellent l'une l'autre. Il ne nous reste plus beaucoup de chemin à faire pour utiliser maintenant une fonction qui s'appelle elle-même : une fonction *récursive*.

Attention : dans les exercices ci-dessous, celle ou celui qui écrit une boucle va au coin!

- 6. **Première fonction récursive - empilements5.** La fonction qui effectue le travail d'empilement de cubes est la fonction `f0`. Comme vous le voyez cette fonction ne contient pas de boucle. La ligne où la fonction `f0` appelle la fonction `f0` dans son propre code est *l'appel récursif*.
- 7. **Suite - empilements5.** Déplacez l'appel récursif *avant* l'empilement. Que se passe-t-il? Expliquez!
- 8. **Suite - empilements5.** Modifiez le code afin d'empiler un cube bleu de gauche à droite, puis un cube vert de droite à gauche, avec un seul appel récursif!
- 9. **A vous - empilements6.** Sur le modèle de l'exercice précédent, écrivez une fonction récursive mais qui fasse une récurrence descendante, de la position 9 à la position 0 en empilant des cubes. Puis modifiez le code afin d'obtenir un empilement de droite à gauche, suivi d'une seconde ligne de gauche à droite (toujours un seul appel récursif.)
- 10. **remplissage - empilements7** La fonction `f0` de l'exercice précédent, dans sa première version permettait de remplir toute une ligne. Pouvez-vous créer une deuxième fonction qui va appeler `f0` pour remplir une ligne, puis qui va s'appeler elle-même afin de remplir la ligne du-dessus? Si cela fonctionne, tout l'écran sera rempli par récurrence. Attention, il faut s'arrêter à un moment quand les lignes deviennent trop hautes (vous pouvez avoir besoin d'un compteur).