Projet python : pavage avec les triminos

On va dessiner, en turtle, la solution du pavage du carré avec les triminos.

0 Consignes de rendu

Ce projet se fait seul(e) ou à deux. Il doit être rendu pour le 3 novembre. La note sera validée par une soutenance individuelle de 10 minutes (qui aura lieu un peu après le rendu).

Pour le code en lui-même, la clarté et la lisibilité du code sont importantes pour l'évaluation. Et les commentaires. Et les commentaires aussi. Sans compter les commentaires.

Les fonctions devront toutes avoir une « docstring » indiquant le format des arguments en entrée, le format de la sortie (s'il y a une valeur retournée), ainsi qu'une brève description de son fonctionnement.

Si vous souhaitez modifier la spécification des fonctions demandées, demandez à votre professeur en justifiant soigneusement votre choix.

La soutenance se fera avec votre professeur(e), sous forme de questions auxquelles l'élève devra répondre. La note finale sera donc individuelle.

1 Généralités et conventions

Vous pouvez choisir entre deux versions : la version « simple » avec un trou forcément dans un coin, et la version « normale » avec le trou placé n'importe où. La version simple sera notée sur 17 et la version normale sur 20 (la différence au niveau du barème est dans la dernière fonction).

Vous devez choisir comment vous allez « coder » la position du trou. Il faudra bien indiquer ce choix dans votre code (voir plus bas).

Exemples : par des coordonnées turtle, par des coordonnées en « comptant » lignes et colonnes, par une chaîne type `"hg"` pour « Haut Gauche » (dans la version simple), ...

2 L'énoncé (5)

Écrire une fonction `generetrou` qui prend en argument un entier `n` et renvoie une position aléatoire du trou dans le carré de 2^n par 2^n, selon votre spécification choisie plus haut. Vous indiquerez soigneusement dans la docstring de cette fonction quelle convention du codage du trou vous avez choisie. **Barème : sur 2**

Écrire une fonction `dessineprobleme` qui prend les arguments suivants : `x,y,n,a,trou`. Elle va « dessiner » le problème du pavage : le carré de côté 2^n et le carré trou (en rouge). **Barème : sur 3**

- `x` et `y` sont les coordonnées (turtle) du coin en bas à gauche du carré à dessiner (cela permet de « placer » le problème),
- `n` est la « taille » du problème. Il faut donc tracer un carré de côté $2^n \times 2^n$.
- `a` est la longueur, en pixels, d'un carré « unité » (cela permet d'adapter l'échelle)

— `trou` est la position du trou d'après la convention choisie. Selon cette convention, cela peut être en plusieurs arguments (par exemple si vous avez un système de coordonnées, vous pouvez avoir deux arguments `xtrou` et `ytrou`)

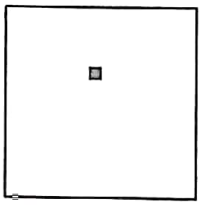

FIGURE 1 – Tracé du problème avec $n = 4$ (donc un carré de côté 16), $a = 10$ et le trou dessiné en rouge (version normale).

3 Le cas de base (4)

Pour rappel, le cas de base ($n = 1$), c'est un carré de 2×2 où il faut dessiner un trimino là où il n'y a pas le trou.

FIGURE 2 – Exemple de tracé du cas de base avec le trou en haut à droite

Écrire une fonction `trimino` qui prend pour arguments `x,y,a,trou`. Elle va résoudre le cas de base ($n = 1$) du problème du pavage, c'est-à-dire le cas où on doit placer un trimino. **Barème : sur 4**

— `x` et `y` sont les coordonnées (turtle) du coin en bas à gauche du carré à paver,
— `a` est la longueur, en pixels, d'un carré unité,
— `trou` est la position du trou d'après la convention choisie. Comme dans la partie précédente, cela peut être plusieurs arguments (le préciser dans la docstring).

Le trimino sera coloré d'une couleur aléatoire (vous pouvez par exemple écrire une fonction qui renvoie une couleur aléatoire et l'appeler dedans).

Indication (à suivre ou non) : écrire une fonction `triminosimple` qui dessine un trimino là où se situe la tortue, avec l'orientation qu'elle a à ce moment. Bien préciser dans la docstring comment le trimino est alors dessiné selon la position et l'orientation de la tortue (par exemple, la tortue est placée au coint en haut à gauche et le trou est en bas à gauche, ou autre situation qui vous convient). Ensuite, dans la fonction `trimino`, on peut placer la tortue au « bon » endroit avec la bonne orientation et appeler cette fonction (on peut utiliser les fonctions `goto` et `setheading`).

4 La fonction principale (10)

Écrire LA fonction récursive `solutionprobleme` qui dessine les triminos dans le carré de $2^n \times 2^n$.
Barème : sur 10 (version simple : **Barème : sur 7**)

Elle prend les arguments suivants : `x,y,n,a,trou`

- `x` et `y` sont les coordonnées (turtle) du coin en bas à gauche du carré à paver,
- `n` est la taille du problème : un carré de 2^n de côté.
- `a` est la longueur, en pixels, d'un carré unité,
- `trou` est la position du trou d'après la convention choisie. Comme dans la partie précédente, cela peut être plusieurs arguments (le préciser dans la docstring).

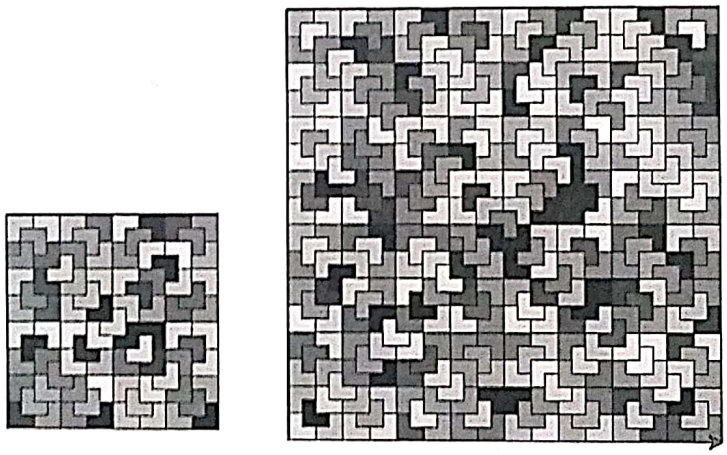

FIGURE 3 – Le problème et sa solution, pour $n = 4$ (carré de côté 16) et $n = 5$ (carré de côté 32).