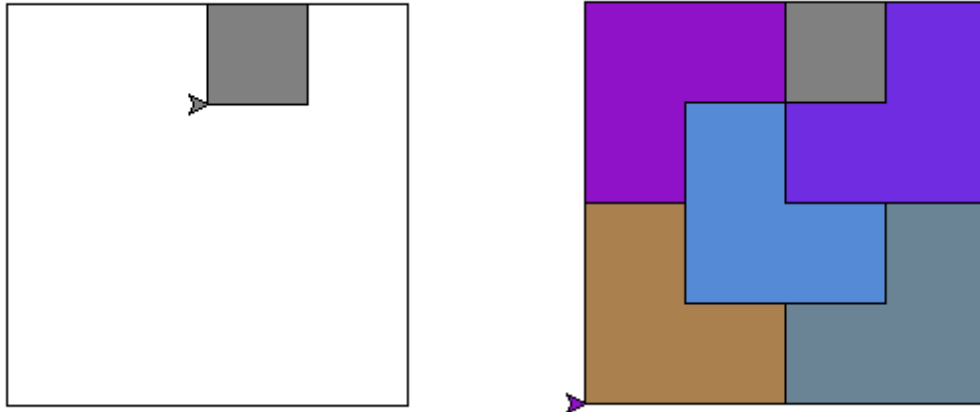


# Triminos

Problème :

On sait que  $2^n * 2^n - 1$  est un multiple de 3

De ce fait si nous construisons un cadrillage de type  $2^n * 2^n$  et que nous enlevons un carré, il nous est possible de remplir le reste du cadrillage avec des triminos, c'est à dire une pièce angulaire formée de 3 carrés :



Le problème est de programmer ceci à l'aide de python et de Turtle.

Nous allons procéder par récurrence.

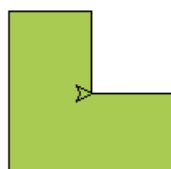
Analysons linéairement le code pour comprendre comment cela peut fonctionner:

Commençons par exposer le problème :

Nous créons une fonction qui dessine un grand carré de type  $2^n * 2^n$  avec  $n$  donné et comme échelle de taille (en pixel)  $a$ .

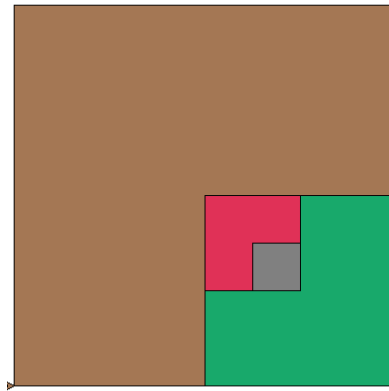
Mais avant cela il nous faut déterminer les coordonnées d'un trou dans le grand carré. Soit on le donne soit il est déterminé aléatoirement. L'outil qui trace le trou sera toujours situé en bas à gauche de celui ci au début et a la fin de son traçage, il faut donc choisir des valeurs de coordonnées aléatoires entre 0 (qui correspondra à la gauche du carré pour  $x$  et au bas du carré pour  $y$ ) et  $2^n - 1$  (qui correspondra à la droite du carré pour  $x$  et au haut du carré pour  $y$ ). Enfin nous pouvons intégrer cette fonction à celle qui dessine le problème, il faudra indiquer au programme que le trou se trouve aux coordonnées données (ou créées) en ajoutant ces coordonnées à la position du carré dans le plan et en indiquant l'échelle, donc en multipliant par  $a$ .

Pour nous mettre dans le bain nous allons créer une fonction simple permettant de tracer un trimino dans le sens où se trouve la tortue (le module python qui permet de dessiner). On indique simplement sa taille en pixel. Voici à quoi ressemble le trimino :

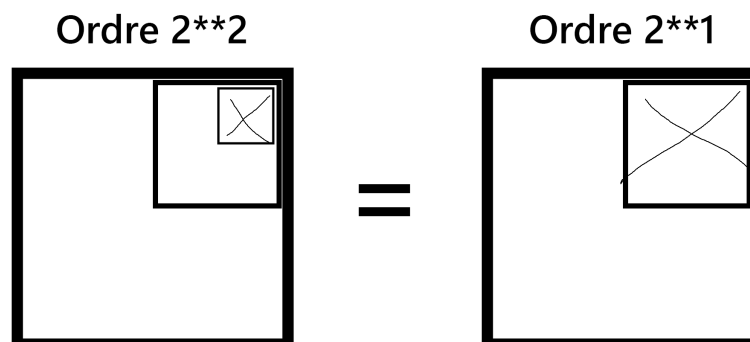


La suite du programme s'annonce un peu plus corcée. Il nous faut trouver un moyen de remplir le carré de triminos peu importe où se trouve le trou et peu importe la taille du cadrillage. Pour cela j'ai décidé de fonctionner par une double récurrence.

Premièrement j'ai constaté que l'on pouvait remplir le carré avec des triminos chacun de taille doublée par rapport à l'autre :



Pour ce faire il faut simplement placer un trimino de façon à ce que le trou + le trimino puissent être considéré comme un "nouveau trou" dans un cadrillage d'ordre inférieur :

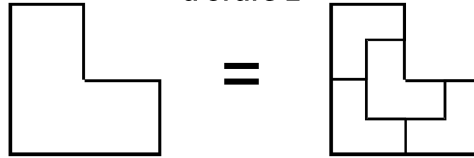


Pour cela il faut vérifier par le calcul l'emplacement de la tortue (qui sera toujours en bas à gauche du trou) par rapport au carré pour savoir où se trouve le trou et donc comment dessiner le trimino : Si la ligne ou la colonne du cadrillage sur laquelle se trouve la tortue est paire ou impaire alors nous savons où se trouve le trou et comment tracer le trimino pour former un nouveau trou. Cette histoire de pair et impair s'inverse dans le seul cas d'un cadrillage de  $2 \times 2$ . il suffit donc d'ajouter à l'équation un nombre qui sera pair tout le temps sauf dans le cas où le côté vaut 2 :  $2^{**n}/2$

Une fois ceci fait on a juste à faire se déplacer la tortue en fonction d'où se trouve le trou et de tracer un trimino de même taille que celui ci. On rappelle ensuite toute cette même fonction jusqu'à ce que le dernier trimino soit de la taille du carré (et donc quand celui ci sera rempli).

Tout cela peut sembler inutile mais une fois que nous avons codé cela il nous suffit de considérer que chaque Gros trimino peut être construit à partir de triminos plus petits :

### Exemple de trimino complexe d'ordre 2



J'appelle ces  
des triminos complexes.

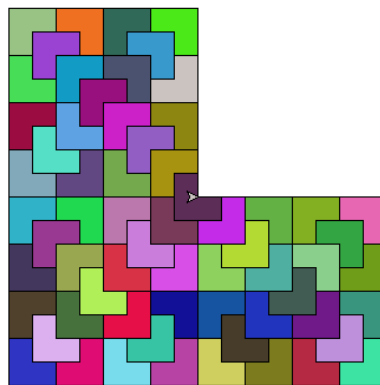
triminos de triminos

C'est là que se trouve notre seconde récurrence. Nous allons créer une fonction qui construit des triminos complexes. Il nous suffit de donner son ordre et son échelle.

L'ordre correspondra en quelque sorte au nombre de triminos qui le constitue.

Par exemple dans l'image précédente, un trimino d'ordre 2 sera constitué de 4 triminos simples, pareillement un trimino d'ordre 3 sera constitué de 4 triminos d'ordre 2 chacun formé de 4 triminos simples et ainsi de suite.

Exemple d'un trimino Complexe d'ordre 4



Une fois que ce programme est finalisé il ne nous reste plus qu'à lier les deux : on fait en sorte qu'au lieu de tracer un "plus grand trimino" il tracera un trimino complexe d'ordre supérieur. Pour connaître son ordre par rapport à sa taille il faut utiliser le fait que sa taille est constamment une puissance de 2 donc on peut utiliser le logarithme :

La taille du grand trimino / La taille d'un trimino simple nous donne un rapport qui correspond à une puissance de 2 donc on utilise le module Math et le logarithme pour connaître l'ordre de cette puissance (on ajoute 1 car  $2^0$  correspond à mon ordre 1).

Et voilà. Le programme fonctionne, par une double récurrence, un carré de type  $2^n * 2^n$  sera rempli de triminos pour tout n.

Voici un exemple de carré de côté 16 et de côté 32 :

