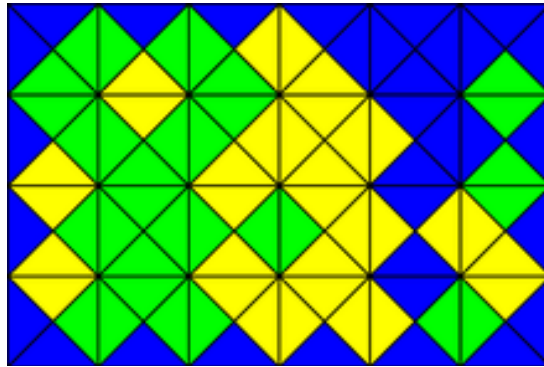


## Projet - Programmation parallèle

### MacMahon Squares



### MacMahon

MacMahon Squares est le nom donné à un puzzle récréatif de correspondance des bords publié pour la première fois par Percy MacMahon en 1921, où 24 tuiles de 3 couleurs uniques sont organisées les uns à côté des autres en associant les couleurs pour créer une grille de 4 par 6. De tels puzzles de pavage ont plusieurs variantes, qui sont déterminées par des restrictions sur la manière de disposer les 24 carrés.

Au début du jeu, la partie commence sur un plateau vierge de taille 4x6 (taille par défaut) et 24 tuiles, ayant chacun une couleur sur chaque bord. Ces couleurs sont : bleu, jaune et vert. Le but est de placer ces tuiles sur la grille, en faisant en sorte que la couleur d'un bord soit le même que celui du carré adjacent. Chaque face de chaque tuile est donc posée à côté d'une face d'une même couleur. Le jeu est terminé quand la grille est remplie avec toutes les tuiles, correctement placés.

**Attention** : Nous allons introduire une variante au jeu MacMahon, exigeant qu'une fois que le plateau est entièrement rempli, la couleur de la bordure du plateau soit la même. Comme le montre l'image ci-dessus, la couleur de la bordure du plateau est la même et bleu.

Un site internet expliquant le jeu MacMahon :

<http://math.en.jeux.chez-alice.fr/mosaiques/macmahoncarre/macmahoncarre.htm>

### Objectif

Vous devez écrire un programme en C++ qui permette de résoudre (remplir correctement le plateau) une partie de jeu MacMahon via un algorithme de type *backtracking* (retour sur trace). Le *backtracking* est un algorithme qui permet de trouver toutes les solutions à un problème donné. L'algorithme construit progressivement des candidats aux solutions et abandonne un candidat dès lors qu'il ne satisfait pas la solution.

Nous jouerons dans une variante du jeu MacMahon ou

Votre programme doit prendre en entrée un fichier texte décrivant une partie (plateau) et devra afficher la solution de cette partie.

Exemple de fichier texte décrivant une partie de jeu :

```
6 6
B B G G
G B G R
G B R G
....
```

La première ligne indique la taille du plateau (nombre de colonne et nombre de ligne) et les autres lignes décrivent les 24 pièces du jeu.

Quelques sites internet expliquant l'algorithme de *backtracking* :

<https://medium.com/@remyc/art-of-backtracking-9e0b8dc70cf2>

<https://igm.univ-mlv.fr/~dr/XPOSE2013/sudoku/backtracking.html>

## Travail à réaliser

1. Modéliser et implémenter le jeu MacMahon
2. Implémenter un algorithme de *backtracking* séquentielle
3. Implémenter un algorithme de *backtracking* parallèles en utilisant la librairie `thread` de C++11. L'algorithme peut être paralléliser à différents niveaux, vous devez implémenter au minimum deux variantes dont une variante utilisant un `thread pool`.

## Rendu du travail

Vous devrez fournir les code sources en C++ du projet ainsi qu'un rapport. Ce rapport devra contenir :

1. Un manuel d'utilisation de votre programme

2. Les explications concernant vos différentes implémentations des algorithmes de *backtracking*
3. Les résultats et/ou courbes de comparaisons entre les différentes variantes de *backtracking*
4. Pour chaque comparaison, vous donnerez une interprétation des résultats obtenus.

Le rapport ne devra pas excéder 3 pages.

Il est fortement conseillé de prendre le temps de la réflexion avant d'arrêter un choix pour vos structures de données. En particulier, les structures de données choisies devront permettre une mise en œuvre aisée et efficace de résolution de *backtracking*.

Durant la phase de débogage, je vous invite à compiler en utilisation les options -Wall et -g3 du compilateur g++ et d'utiliser alternativement un débogueur comme gdb et un programme vérifiant les accès à la mémoire comme valgrind.

Une fois votre programme au point, il est conseillé de le compiler avec l'option d'optimisation -O3 du compilateur g++ afin d'obtenir un exécutable le plus performant possible. Dans le cas de figure où vos expérimentations sont effectuées sur un ordinateur portable, il est important de s'assurer que la politique de gestion de l'énergie est réglée sur la puissance maximum afin de ne pas biaiser les comparaisons.

**Attention** : de nombreuses ressources et programmes (de qualité variable) sont disponibles très facilement, notamment sur Internet. Vous pouvez bien évidemment les consulter, mais gardez bien à l'idée que c'est un travail personnel : l'enseignant responsable sera particulièrement vigilant sur ce point et prendra les sanctions adéquates en cas de détection.