

# Les 7 merveilleuses merveilles du merveilleux monde des 7 merveilleuses couleurs

Aurèle Barrière & Nathan Tomasset

8 mars 2016

## Intoduction

Nous présentons dans ce projet une implémentation du jeu des 7 couleurs dont les règles seront décrites plus bas. L'enjeu est d'implémenter en C le jeu, puis de créer des stratégies pour jouer.

## Règles du jeu

Le jeu se présente comme un tableau carré de 30 case de côté (ce nombre est paramétrable dans notre implémentatoin). Initialement, il est rempli aléaoirement avec 7 “couleurs” (ou lettres). Dans deux angles opposés, il y a une huitième et une neuvième couleurs : celles des joueurs. Tour à tour, les joueurs choisissent une couleur parmi les 7. Toutes les cases de cette couleur et adjacente à la couleur du joueur (ou adjacente à une case qui vient de changer de couleur à ce tour par cette méthode) prennent désormais la couleur du joueur. Le but est de posséder plus de la moitié du plateau.

## Organisation du projet

## Voir le monde en 7 couleurs

### 2.1

Pour initialiser le tableau de manière aléatoire, on parcourt la matrice (initialisée avec des 0), et on remplace chaque coefficient par une des 7 couleurs.

Pour obtenir ce nombre, on importe le module `stdlib`, on initialise le générateur pseudo-aléatoire avec le temps (`srand(time(NULL));`) et on prend le résultat de la fonction `rand()` modulo le nombre de couleurs.

Cependant, il faut bien initialiser l'aléatoire au début de la fonction principale, et pas à chaque partie. En effet, dans la suite on lancera plein de parties en même temps, qui seront donc identiques si lancées à moins d'une seconde d'intervalle.

Enfin, `rand()` modulo un nombre de couleurs ne produit pas un résultat avec une probabilité parfaitement uniforme. Cependant, vu le petit nombre de couleurs, ce n'est pas un problème significatif.

## 2.2

On crée une fonction `void update_board(char player, char color, char * b)`. Elle prend en argument un joueur `player`, une couleur jouée `color` et un plateau `b`.

Elle consiste à parcourir la matrice, et si on trouve une case qui est de la couleur jouée et à côté d'une case de la couleur du joueur, on la remplace et on indique dans une variable qu'il y a eu un changement. S'il y a eu changement, il faut recommencer le parcours de la matrice.

On pose  $n$  la taille du tableau. Dans le pire cas, il faut réappliquer  $n$  fois le parcours de matrice (à chaque parcours, il y a un changement). On a donc une complexité en  $\mathcal{O}(n^3)$  dans le pire cas.