



<https://gabrielecirulli.github.com/2048/>

Attention nous considérons que le jeu ne se termine que lorsque plus aucun mouvement n'est possible.

Niveau 1:

Proposer une Version java graphique du Jeux (utilisant pour l'interface utilisateur soit les touches, soit un click de souris dans une zone spécifique au 4 directions, soit une gesture qui indique une des 4 directions .

Couleur et dynamique des mouvements a votre gout (attention au mauvais gout, demandez à quelqu'un si vous êtes daltonien).

Le tirage de 2 ou 4 est 2 pour les deux premières valeurs et à 90% des 2 et 10% des 4. La position de la nouvelle valeur est tirée de façon uniforme sur les case vides.

[test manuel si j'arrive à jouer c'est ok !]

Niveau 2: (la moyenne) Le logiciel permet de définir sur la ligne de commande les options et paramètres suivants :

-s fichierdesauvegarde -r fichierdereplay
-a fichieraleatoire -n fichieraleatoire

la commande -s permet de sauvegarder la partie dans un fichier pour être rejouée avec l'option -r.

le format de fichier de sauvegarde et de replay est le suivant

des lignes de commentaires commençant par # puis

deux lignes de la forme

2VH

puis une suite de ligne de la forme NVHD

Ou N est soit le caractère '2' soit le caractère '4'
Ou V est un des caractère '1' à '4' pour la position Verticale
Ou H est un des caractère '1' à '4' pour la position Horizontale
Ou D est un des caractères 'N','S','E','W', une direction Nord (haut), Sud(bas), Est (droite), Ouest(Gauche, West).

Si le jeu se termine avant la fin de fichier afficher un message d'erreur (popup svp) puis quitter le programme avec une valeur d'erreur égal à 12. [test automatique].
Si le jeu est encore possible donner la main au joueur dans l'interface graphique pour qu'il puisse continuer l'enregistrement si il a lieux.

Gerez le fait que le fichier de replay et de sauvegarde soit le même fichier.

Le fichier fournis par l'option -a contient une suite de bits 0 ou 1 indiquant si la prochaine valeur du jeu est 2=0 ou 4=1 soit un fichier produit par l'option -n.

Quand vous faites un replay il est possible que le jeu soit encore possible et donc les valeurs du tirage suivants sont fournies par le fichier format -a ou -n fournit.

[test avec mon fichier aléatoire à moi, je lance -s toto -a fichieraleatoireamoi,
je joue trois fois, puis j'arrête,
en suite je regarde le contenu du fichier toto si il est égal au mien c'est bon !]

Niveau 3: Le solveur

Changement de la définition voir le post sur le « news forum ».

Le solveur doit fournir le meilleur jeu pour une suite aléatoire définie par le générateur aléatoire standard de java et la graine (seed) à utilisée est 14071789 (date célèbre).

Il faut donc avoir un fonction qui fournisse la position et la valeur de la nouvelle piece d'une façon uniforme pour tous.

Ainsi nous ferons appel deux fois au générateur aléatoire une fois avec un entier compris entre 0 et 99 si la valeur est inférieur ou égal à 10 c'est un 4 sinon c'est un deux.

En suite le nombre de case vide NCV sera calculé puis un entier compris entre 0 et 999999 nous donneras la position de la case vide avec valeur modulo NCV qui est remplacée par la nouvelle valeur. Ainsi le générateur aléatoire tire toujours la même valeur quelque soit le nombre de cases vides.

r= new Random(long seed) Creates a new random number generator using a single long seed.

r.nextInt(int n)

Returns a pseudorandom, uniformly distributed int value between 0 (inclusive) and the specified value (exclusive), drawn from this random number generator's sequence.

Une petite remarque sur le générateur aléatoire il ne sais pas revenir en arrière....

[test et bien moi je ne vais pas loin j'atteint pas 2048 avec mon solver, le meilleur d'entre vous auras une prime ;) qui ne pourras pas faire monter sa note au dessus de 20. les 5 suivants une prime aussi ...] Celui qui atteint 32688 a triché. Par contre 80 de plus c'est ok.

Pour compter les points il suffit de faire la somme des valeurs des pieces affichés.

Niveau 4: En supposant que le jeu oppose un joueur humain à l'ordinateur (qui n'a pas de stratégie ...) utiliser l'algorithme MinMax pour et le pruning alpha-beta voire le document suivant :

<http://blog.datumbox.com/using-artificial-intelligence-to-solve-the-2048-game-java-code/>

Il vous est demandé de ne pas réutiliser directement le code de [datumbox.com](http://blog.datumbox.com/using-artificial-intelligence-to-solve-the-2048-game-java-code/) comme les énumérations ne sont pas bien utilisées (euphémisme) et que la transmission d'information a travers le code se fait en utilisant des dictionnaires !!!!

Une nouvelle fonction heuristique d'évaluation plus efficace sera appréciée.

Niveau 5:

Au fait il faut une grille 5x5 ou 6x6 ou

;)