

# Licence Informatique 1<sup>e</sup> année Algorithmique et Programmation

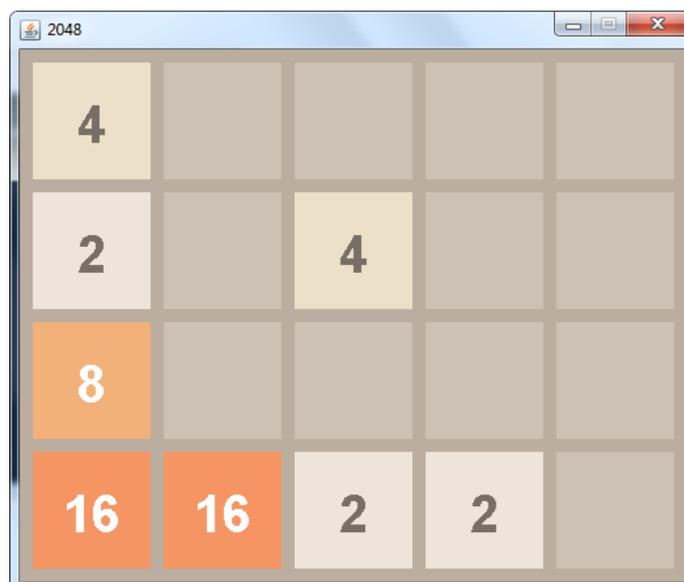
Projet 2015-2016

## 1 Organisation

Le projet est à réaliser en binôme. En fin de semestre, les binômes présenteront leur travail aux encadrants de TP. Lors des présentations, chaque binôme rendra un rapport qui présentera le programme réalisé (structures de données utilisées, principe des algorithmes implémentés, ...), les résultats obtenus, les problèmes rencontrés, les solutions apportées et tout élément permettant d'évaluer le travail réalisé. Chaque binôme devra également, au moment des soutenances, envoyer le code des programmes écrits aux responsables de TP (*clement.lecat@u-picardie.fr*, *yu.li@u-picardie.fr* et *frederic.furst@u-picardie.fr*).

## 2 Sujet

Le projet consiste à programmer le jeu 2048, dont une version est disponible en ligne sur la page <http://gabrielecirulli.github.io/2048/>. Le programme réalisé doit permettre de jouer au jeu tel qu'il se présente sur le site Web. Il doit également permettre au joueur de choisir au démarrage différents paramètres, au minimum la taille de la grille (pas forcément carrée) et le nombre à atteindre pour gagner (2048, moins, ou plus).



La partie interface graphique n'est pas à votre charge. Une classe Java *Interface2048.java* est fournie sur la page du module. Son mode d'emploi est donné plus bas. Vous devez coder tout le reste.

**Remarque importante :** des programmes de type 2048 écrits en Java peuvent surement être trouvés sur le Web, mais ils seront écrits en Java objet. Aucun projet écrit dans le paradigme objet ne sera pris en compte lors des soutenances.

## 2.1 Fonctionnalités à réaliser optionnellement

Ces extensions du programme de base apportent des points en plus au projet **à condition que les fonctionnalités obligatoires aient été réalisées.**

- Vous pouvez gérer les données liées au jeu à l'aide de fichiers. Ces données peuvent concerner les parties (sauvegarde d'une partie en cours, chargement d'une partie sauvegardée) et/ou les joueurs (sauvegarde des meilleurs scores de chaque joueur, de l'historique des parties de chaque joueur, ...).
- Vous pouvez ajouter des éléments supplémentaires au jeu, comme des cases obstacles, une limite de temps, etc. La difficulté du jeu peut aussi être modulée en fonction de la façon dont sont générés les numéros (on peut par exemple générer autre chose que des 2 ou des 4).
- Vous pouvez ajouter de la musique<sup>1</sup>.

## 2.2 Démarche générale pour programmer le jeu

La première chose à faire est de comprendre le jeu, en jouant. Il faut ensuite bien identifier les données à manipuler : quelles données sont nécessaires pour représenter la grille du jeu et les contenus des cases ? Une fois les données à représenter identifiées, vous devez choisir des structures de données à utiliser. Par exemple, il paraît incontournable d'utiliser un tableau à deux dimensions pour représenter la grille. Mais on peut choisir de coder le contenu de chaque case avec des nombres, ou avec des enregistrements ou autre chose.

Une fois les structures de données choisies, vous pouvez déjà écrire le code Java correspondant à la déclaration de vos structures et à leur initialisation. À ce stade, vous pouvez déjà utiliser l'interface graphique pour afficher l'état initial du jeu.

L'étape suivante est d'écrire l'algorithme qui fait tourner le jeu. Le mieux est d'y aller étape par étape. Par exemple, commencez par écrire un algorithme (sous forme d'une fonction par exemple) qui génère aléatoirement des numéros sur les cases libres. Gérez ensuite les déplacements des numéros sur la grille. Ajoutez ensuite la gestion de la fusion des numéros. Finalement, on peut écrire l'algorithme général du jeu avec la détection de la fin du jeu (chiffre cible atteint ou impossible de continuer à jouer).

---

1. Voir le tutoriel <http://docs.oracle.com/javase/tutorial/sound/>

Pour gérer la vitesse du jeu, on peut utiliser l'instruction `try{Thread.sleep(n);} catch(InterruptedException e){}` qui met le programme en attente `n` millisecondes.

### 3 L'interface d'affichage

Du code est fourni pour l'affichage du jeu. Il est obligatoire de l'utiliser. Pour l'utiliser, placez simplement le fichier `Interface2048.java` dans le répertoire où se trouve votre programme, vous pourrez alors utiliser dans votre programme les instructions données plus bas. Il est inutile de lire le contenu de cette classe, et encore moins de le comprendre, pour l'utiliser. Les explications qui suivent suffisent pour l'utiliser.

#### 3.1 Les instructions disponibles pour utiliser l'interface

**Instruction pour créer une interface:** `Interface2048 toto = new Interface2048(1, h)` où `1` et `h` sont respectivement la largeur et la hauteur de la grille (en nombre de cases).

**Instruction pour changer le contenu d'une case :** `toto.modifieCase(x,y,val)` où `toto` est la variable contenant l'interface créée, `x` et `y` sont les abscisses et ordonnées de la case (de type *int*) et `val` est la valeur de type *int* à afficher dans la case. Si `val` vaut 0, la case est affichée vide.

**Instruction pour savoir quelle touche l'utilisateur a tapée:** `toto.toucheTapee()` renvoie un *int* qui vaut 1 pour la flèche gauche, 2 pour la flèche droite, 3 pour la flèche bas et 4 pour la flèche haut.

**Instruction pour afficher un message dans une boîte de dialogue:** `toto.afficheMessage(m)` où `toto` est la variable contenant l'interface créée et `m` la chaîne de caractères à afficher.