

# Projet PP1 2015 : 2048

## 1 Présentation

Dans le cadre de ce projet, on s'intéresse au développement du jeu 2048 [1]. Il se joue sur une grille de 4x4 cases avec des tuiles de valeurs variées et toujours en puissance de 2. L'utilisation des touches fléchées permet de déplacer les tuiles dans la grille selon 4 directions : la gauche, la droite, le haut ou le bas. Lorsque deux tuiles de même valeurs entrent en "collision", celles-ci fusionnent en une nouvelle valeur égale à 2 fois leur valeur. Par exemple, lorsque deux tuiles de valeur 2 fusionnent, la nouvelle tuile a pour valeur 4. Lorsqu'un mouvement est achevé, une nouvelle tuile de valeur 2 ou 4 se place aléatoirement sur une des cases libres de la grille. Le but du jeu est d'arriver à construire une brique de valeur la plus grande possible (2048, 4096, 8192,...).

## 2 Travail demandé

Le but de projet est de fournir une application permettant dans un premier temps de jouer à 2048. Cette application devra s'appuyer sur l'interface de la grille qui est décrite dans le fichier `grid.h` présent dans ce répertoire. A l'issue de la première partie, chaque groupe devra faire une relecture critique (mais surtout constructive) du code de 3 autres groupes. Dans une seconde partie, lorsque votre jeu sera fonctionnel, vous devrez implémenter des stratégies permettant de jouer et de gagner le plus souvent possible.

### 2.1 Première partie

Pour cette partie, chaque groupe doit fournir les sources permettant de générer une bibliothèque `libgrid.a` (implémentant les fonctions décrites dans `grid.h`) ainsi qu'un exécutable permettant de jouer à 2048 dans la console.

Ce code sera à déposer dans le répertoire `/net/stockage/V1/` au plus tard le **lundi 23 mars 8h** sous forme d'une archive `<grp>_<nom1>_<nom2>_<nom3>.tar.gz` (où `<grp>` désigne le numéro du groupe et `<nom1>`, `<nom2>` et `<nom3>` désignent les noms des membres du groupe (par exemple `A1_durand_frederique_kirman.tar.gz`).

## 2.2 Relecture critique

A l'issue de cette première phase, chaque groupe devra relire le code produit par 3 autres groupes. Pour chaque groupe à relire, vous aurez à remplir un formulaire d'évaluation. Ce formulaire d'évaluation vous sera fourni un peu plus tard dans le semestre. Une fois rempli, ces formulaires seront à envoyer par mail à votre chargé de TD au plus tard le **lundi 30 mars 8h**.

## 2.3 Deuxième partie

Dans cette deuxième partie, chaque groupe doit proposer un exécutable permettant de jouer à 2048 en mode graphique ainsi que 2 stratégies : une stratégie rapide (capable de jouer une partie en moins de 10s) et une stratégie plus gourmande (capable de jouer une partie en moins de 2min). Une stratégie est une structure qui implémente l'interface décrite dans `strategy.h`. Chaque stratégie devra prendre la forme d'une bibliothèque dynamique contenant une fonction dont le nom est le même que celui de la bibliothèque dynamique. Par exemple dans la bibliothèque `libA1_durand_frederique_kirman_fast.so` se trouve la fonction `strategy A1_durand_frederique_kirman_fast()` ;.

Les stratégies sont à déposer respectivement dans `/net/stockage/pp1/strategies/fast` et `/net/stockage/pp1/strategies/efficent` au plus tard **lundi 13 avril 8h**. Un classement des différentes stratégies sera mis en ligne dans la semaine qui suivra.

Pour implémenter votre interface graphique vous pouvez utiliser par exemple la bibliothèque SDL [3] ou la bibliothèque MLV [2].

L'intégralité du projet sera à fournir lors de la soutenance qui se tiendra pendant la séance de TP de la semaine 17 (**mardi 21 ou mercredi 21 avril** suivant les groupes).

## 2.4 Organisation

Le projet est étudié et réalisé en trinôme mais la notation est individuelle. L'enseignant évaluera la contribution de chacun des éléments du trinôme au travail commun. L'enseignant se réservera la possibilité de modifier la composition de chacun des trinômes. Afin de permettre un travail profitable, il est conseillé de ne pas créer de groupes avec des niveaux trop différents.

Le projet doit être réalisé à l'aide des différents outils disponibles dans votre environnement de développement, à savoir "profilage de code", "débogage", "documentation automatique de code", "gestion de versions" (vous pouvez utiliser par exemple le site savane du CREMI : <https://services.emi.u-bordeaux1.fr/projet/savane/>).

Deux jours avant chaque TD consacré au projet, une brève synthèse (1/2 page environ) des travaux effectués depuis la séance précédente sera transmis (mail ou papier) à l'enseignant. On précisera les points que l'on désire plus

particulièrement aborder avec l'enseignant chargé de suivre le projet. Chaque document contiendra la date et les noms du trinôme.

## 2.5 Soutenance

La présentation finale du projet se fera en salle de TP. Cette présentation se fera autour d'une démonstration et d'un compte-rendu écrit présentant :

- les objectifs atteints et ceux qui ne le sont pas ;
- l'exposé d'un problème technique rencontré et sa résolution ;
- quelques exemples montrant que vous avez su éviter des redondances dans votre code.

Il s'agira aussi de bien préciser l'origine de toute portion de code empruntée (sur internet, par exemple) ou réalisée en collaboration avec tout autre trinôme. Il est évident que tout manque de sincérité sera lourdement sanctionné.

## 2.6 Présence

La présence aux TD, à la soutenance et au devoir surveillé terminal est obligatoire. Toute absence injustifiée donnera la note 0 pour l'épreuve concernée. En cas de circonstances exceptionnelles, contacter son enseignant de TD par mail au plus tard le jour de la séance de TD concernée ou le jour du devoir surveillé.

## Références

- [1] 2048 game. <http://gabrielecirulli.github.io/2048/>.
- [2] Bibliothèque graphique mlv. <http://www-igm.univ-mlv.fr/~boussica/mlv/index.html>.
- [3] Simple directmedia layer. <http://www.libsdl.org/>.