

## Projet

### 1 Généralités

Ce projet a pour but la construction d'une plateforme, ou framework, permettant d'implémenter plusieurs jeux proches : il s'agit de jeux de déplacement de blocs avec différents buts et modalités selon le jeu.

Un framework est une sorte de bibliothèque logicielle dont la caractéristique principale est l'usage important de différentes formes de généricité et qui définit un cadre encourageant certaines architectures logicielles. Vous noterez que les jeux ont un certain nombre de similitudes ; si votre framework est correctement architecturé, l'implémentation de chaque nouveau jeu s'en trouvera grandement simplifiée.

L'obtention de ce framework devra se faire à l'aide d'une analyse soignée des différents jeux.

### 2 Réalisation

Votre code doit être proprement conçu et documenté. Les réalisations de tests, de phases de démonstrations sont fortement encouragés. Ne vous lancez pas tout de suite dans l'écriture du code ! Commencez par prendre du temps pour analyser comment ces jeux fonctionnent et quelles sont les briques utilisées, etc.

L'accent doit être mis sur l'architecture logicielle, de sorte que le code soit aisé à lire et modifier ; ce point est très important car il permettra de vérifier vos compétences dans la maîtrise d'un développement orienté objet. Le framework devra avoir été pensé de sorte qu'on puisse sans trop de difficultés imaginer comment implanter d'autres jeux du même type.

On ne vous demande pas d'interface graphique, une interface texte suffira. Si vous avez du temps et que vous voulez quand même en faire une, rappelez-vous qu'une jolie interface collée sur un mauvais projet ne vous fera pas gagner de points et peut même vous en enlever si cela démontre une mauvaise prise en compte des priorités. Commencez donc dans tous les cas, par faire le jeu sous forme d'interface texte. Si vous voulez faire une interface graphique nous vous recommandons la librairie SfmL (cf. page du cours).

### 3 Coté pratique

Le projet doit être réalisé en **binômes**, nous vous en demanderons la constitution très rapidement. Si vous faites le projet seul, vous serez noté comme si vous étiez en binôme. Les trinômes et autres sont bien sûr interdits.

La modélisation est à rendre sur Didel sous format *PDF* d'ici le 1er Décembre, le nom des participants au binôme sera indiqué dans le *PDF* et dans le nom du fichier.

Vous aurez le droit de modifier cette modélisation.

Le projet est à soumettre, sous forme d'une archive (tar ou zip) contenant le nom des participants au binôme. (Pas d'archive rar). Elle devra contenir :

- les sources de votre programme
- un fichier nommé **README** qui indique comment on se sert de votre programme ;

- un rapport au format *PDF* de quelques pages expliquant concisément les parties traitées, les problèmes connus, et les pistes d’extensions que vous n’auriez pas encore implémentées. Naturellement, il contiendra un diagramme de classes. Vous imprimerez ce rapport pour le jour de la soutenance. La date de rendu du projet vous sera communiquée ultérieurement.

## 4 Présentation des jeux

### 4.1 Travail à réaliser

Vous devez modéliser l’ensemble des jeux. Pour la réalisation, vous devez faire deux jeux dont 2048 et 2 variantes de 2048.

De plus, pour tous les jeux : vous devez programmer, en plus d’un joueur humain, un joueur “robot stupide” qui à chaque tour, jouera un coup autorisé (si on peut encore jouer) : le premier que le programme trouve. Il ne s’agit donc pas d’intelligence artificielle!

Par ailleurs, pour tous ces jeux, la taille du plateau doit être paramétrable.

### 4.2 Taquin

Le premier jeu est le taquin. Vous en trouverez l’explication à l’adresse suivante : <https://fr.wikipedia.org/wiki/Taquin>.

Pour préciser un peu l’explication : Le principe est donc qu’on peut déplacer un petit carré à côté d’un trou à l’endroit où il y le trou, un nouveau trou se formant à l’endroit initial du petit carré. Le but du jeu étant de remettre les petits carrés dans l’ordre.

### 4.3 2048 avec variante

Vous pouvez tester le jeu 2048 original à l’adresse suivante : <http://gabrielecirulli.github.io/2048/>.

**Brève explication :** le plateau rectangulaire est composé de cases qui sont soit vides soit portent un nombre : une puissance de deux. Lorsqu’on penche ce plateau suivant l’un de ses bords, l’ensemble des nombres, soumis à la gravité tombent du même coté. Des nombres égaux fusionnent alors pour former un seul nouveau nombre dont la valeur est le double des nombres fusionnés. Avant le prochain mouvement, un nouveau nombre (2 ou 4) apparaît au hasard et ainsi de suite. L’objectif est de jouer le plus longtemps possible, et donc d’obtenir les plus grandes valeurs possibles.

Pour ce jeu, en plus du jeu original, nous vous demandons de proposer au moins 2 options/variantes supplémentaires, par exemple :

- On accepte d’avoir sur le plateau d’autres nombres que des puissances de deux. Par exemple des 3, des 5 etc. Si la fusion reste possible entre égaux, les nombres engendrés par des 3 resteront incompatibles avec ceux engendrés par les 2 etc.
- S’il y a des nombres négatifs, ceux-ci auront la possibilité de détruire un nombre opposé.
- Les tuiles “×2” multiplient par deux la valeur de la tuile avec laquelle elle fusionne. Sur le même schéma on peut imaginer une tuile “diviser par 2”.
- Certaines tuiles “destroy” ont la particularité de supprimer la tuile avec qui elle fusionne quelle que soit sa valeur et de disparaître également.

## 4.4 Sokoban très simplifié

Une explication et une animation du jeu est fournie à la page suivante  
<https://en.wikipedia.org/wiki/Sokoban>

Le but du jeu est pour le personnage de déplacer toutes les caisses sur les buts marqués (points roses). Il faut tenir compte du fait que le joueur ne peut se déplacer que sur des cases vides, qu'il ne peut pousser qu'une caisse à la fois, et que, bien sûr, il ne peut pas déplacer les murs.

**Version à implémenter** Dans la version à implémenter, on place de manière aléatoire un certains nombres de murs, un certains nombre de buts et les emplacements initiaux des caisses et du personnage. On ne vous demande pas de vérifier que le jeu engendré a une solution, ni de prévoir des fichiers de niveaux.