

TP C#14

2048

Rendu

Archive

Vous devez rendre un fichier au format .zip respectant l'architecture suivante:

```
rendu-tp-firstname.lastname.zip
|-- rendu-tp-firstname.lastname/
|   |-- AUTHORS
|   |-- README
|   |-- 2048
|       |-- 2048.sln
|       |-- 2048
|           |-- Program.cs
|           |-- Grid.cs
|           |-- Score.cs
|           |-- Controller.cs
|           |-- Number.cs
|           |-- Tout le reste sauf bin/ et obj/
```

- Bien entendu, *firstname.lastname* correspond à votre login et vous devez le remplacer par votre prénom et votre nom.
- Votre code doit compiler et être lisible.
- Exceptionnellement, l'architecture de rendu peut être modifiée dans le but d'ajouter les bonus présentés ainsi que ceux qu'il vous plaira d'ajouter.

AUTHORS

Ce fichier doit contenir une ligne formatée comme il suit : une étoile (*), un espace, votre login et un retour à la ligne, RIEN DE PLUS.

Voici un exemple (où \$ représente un retour à la ligne et _ un espace):

```
*_firstname.lastname$
```

Notez que le nom du fichier est AUTHORS avec AUCUNE extension. Pour créer simplement un fichier AUTHORS valide, vous pouvez taper la commande suivante dans un terminal:

```
echo "* _firstname.lastname" > AUTHORS
```

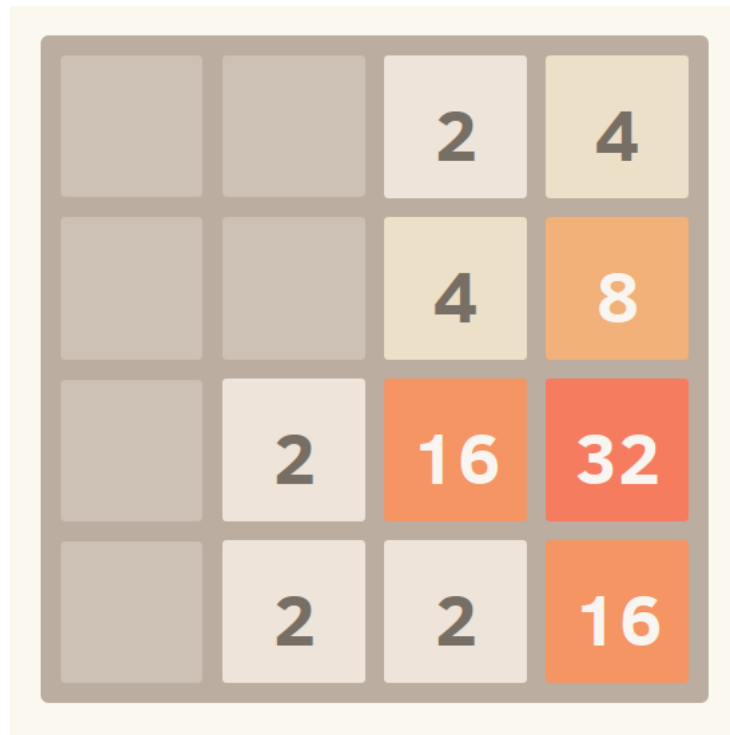
README

Vous devez écrire dans ce fichier tout commentaire sur le TP, votre travail, ou plus généralement vos forces / faiblesses. N'oubliez pas de présenter les bonus que vous implémentez. Un README vide sera considéré comme une archive invalide (malus).

1 Introduction

Vous connaissez sûrement le célèbre jeu 2048 (<https://gabrielecirulli.github.io/2048/>). L'objectif de ce TP sera de le faire en C#. Ce sera un bon exercice de révisions des partiels, c'est un sujet qui est déjà tombé il y a quelques années. Vous serez moins guidés pour ce TP. Nous vous conseillons de réaliser ce TP dans des conditions d'examen. Vous devez respecter l'architecture que vous avez en page 1. Une partie du barème prendra en compte votre bonne utilisation du C#.

Le jeu se joue sur une grille de 4x4 et le but est de faire combiner les tuiles du même nombre pour obtenir le plus grand nombre possible. Le jeu se termine lorsque le nombre 2048 est atteint ou lorsque qu'il n'est plus possible de combiner les tuiles.



2 2048

2.1 Program.cs

Vous devez y instancier toutes les classes (sauf score) et démarrer votre contrôleur.

2.2 Controller.cs

L'utilisateur va devoir interagir avec le jeu et vous allez donc devoir récupérer les touches appuyées par l'utilisateur comme vous avez déjà eu l'occasion de le faire à maintes reprises. Une solution qui vous permet d'avoir plus de modularité et de clarté dans votre code est de créer une classe qui aura une méthode lui permettant d'analyser l'entrée de l'utilisateur et de retourner l'action qui correspond. Vous pouvez également opter pour une énumération pour gérer les différentes actions possibles. Vous pourrez alors effectuer l'action correspondant sur votre grille.

```
1 public class Controller
2 {
3     public enum Action{Up, Down, Left, Right, Quit};
4     public Action GetAction();
5 }
```

2.3 Grid.cs

C'est la plus grosse classe, vous devez pouvoir avec cette classe :

- Déplacer toutes les cases de numéro dans les 4 directions
- Mettre à jour la classe **Score**.
- Vérifier si la partie est terminée. (Bonus)
- Gérer l'affichage de la grille.

Lors du déplacement, les tuiles, qui sont déjà sur le bord de la grille vers lequel vous bougez, ne bougent pas. Et lors des déplacements, vous devez faire apparaître, de manière aléatoire une tuile de valeur inférieure ou égale à 4.

2.4 Score.cs

Ceci devra être une classe statique. Elle contient le score et en bonus le meilleur score.

```
1 public static class Score
2 {
3     /* FIXME */
4 }
```

Le score est comptabilisé de la manière suivante : à chaque fois que vous combinez des tuiles ensembles, vous ajoutez leur valeur au score total (si vous fusionnez 2 tuiles de 2, vous ajoutez 4, si vous fusionnez 2 tuiles de 8, vous ajoutez 16...).

2.5 Number.cs

Cette classe est facultative, vous en aurez très probablement besoin si vous allez plus loin dans votre 2048. Elle peut par exemple stocker le numéro et la couleur de ce numéro à afficher. Comme cela vous n'aurez pas à calculer la couleur du numéro. La couleur du numéro est un bonus.

2.6 Bonus

Libre à vous de faire tous les bonus que vous voulez. Essayer de faire la partie obligatoire dans des conditions d'examen puis améliorez votre 2048 par la suite. Nous attendons de vous beaucoup de bonus!

Quelques exemples :

- Afficher un message de victoire (Quand le carré 2048 est atteint)
- Afficher un message en cas de défaite

- Sauvegarde du score dans un fichier
- Sauvegarde de la grille dans un fichier
- Affichage avec des couleurs

N'oubliez pas de lister les bonus dans votre README !

The code is the law.