INFORMATIQUE
Sequence 5 : Fonctions

TP Mini Projet

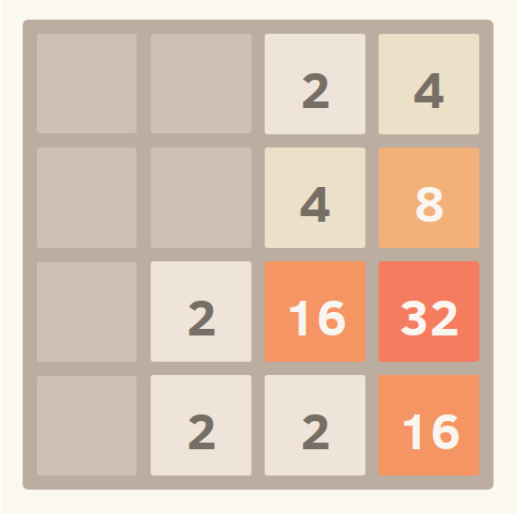
v2025-09-24

IUT d'Annecy, 9 rue de l'Arc en Ciel, 74940 Annecy

Télécharger et décompresser le squelette du projet :

```
wget https://github.com/IUT-GEII-Annecy/squelettes/releases/download/branch-2025/projet_2048.zip
unzip projet_2048.zip
rm projet_2048.zip
```

1 2048



On souhaite développer le célèbre jeu 2048 en C en utilisant des fonctions pour structurer le code. Le but du jeu est de faire glisser des tuiles numérotées sur une grille afin de combiner des tuiles identiques et atteindre la tuile 2048.

1.1 Représentation de la grille

La grille de jeu est une matrice 4x4 d'entiers. Chaque case peut contenir un nombre (2, 4, 8, etc.) ou être vide (représentée par 0).

```
#define SIZE 4
typedef int Grid[SIZE][SIZE];
```

Le projet sera organisé de la façon suivante :

- `jeu_2048.c` : Contient la fonction `main` et la logique principale du jeu.
- `affichage.c` : Contient les fonctions liées à l'affichage de la grille.
- `deplacement.c` : Contient les fonctions pour gérer les déplacements des tuiles.
- `utils.c` : Contient des fonctions utilitaires, comme l'initialisation de la grille et la vérification de la fin du jeu.
- `affichage.h`, `deplacement.h`, `utils.h` : Fichiers d'en-tête correspondants pour déclarer les fonctions.

1.2 Fonctions à implémenter

- **Initialisation de la grille** : Une fonction pour initialiser la grille avec deux tuiles aléatoires (2 ou 4).
- **Affichage de la grille** : Une fonction pour afficher la grille dans la console.
- **Déplacement des tuiles** : Quatre fonctions pour déplacer les tuiles vers le haut, le bas, la gauche et la droite. Chaque fonction doit gérer la fusion des tuiles identiques.
- **Ajout d'une nouvelle tuile** : Une fonction pour ajouter une nouvelle tuile (2 ou 4) dans une case vide après chaque déplacement.
- **Vérification de la fin du jeu** : Une fonction pour vérifier si le joueur a gagné (atteint la tuile 2048) ou perdu (plus de mouvements possibles).

Manipulation 1 : Initialisation et affichage de la grille

1. Implémentez la fonction `void initGrid(Grid grid)` qui initialise la grille avec deux tuiles aléatoires.
2. Implémentez la fonction `void displayGrid(const Grid grid)` qui affiche la grille dans la console.
3. Testez ces fonctions dans le fichier `jeu_2048.c` en initialisant une grille et en l'affichant.
4. Compilez et exécutez le programme pour vérifier que la grille est correctement initialisée et affichée.
5. Faire valider par l'enseignant.

Manipulation 2 : Déplacement des tuiles

1. Implémentez les fonctions de déplacement : `void moveUp(Grid grid)`, `void moveDown(Grid grid)`, `void moveLeft(Grid grid)`, `void moveRight(Grid grid)`.
2. Chaque fonction doit gérer la fusion des tuiles identiques.
3. Testez ces fonctions en effectuant des déplacements sur une grille initialisée.
4. Compilez et exécutez le programme pour vérifier que les déplacements fonctionnent correctement.
5. Faire valider par l'enseignant.

Manipulation 3 : Ajout d'une nouvelle tuile et vérification de la fin du jeu

1. Implémentez la fonction `void addNewTile(Grid grid)` qui ajoute une nouvelle tuile (2 ou 4) dans une case vide.
2. Implémentez la fonction `int checkGameOver(const Grid grid)` qui vérifie si le joueur a gagné ou perdu.
3. Testez ces fonctions en les intégrant dans la boucle principale du jeu.
4. Compilez et exécutez le programme pour vérifier que l'ajout de nouvelles tuiles et la vérification de la fin du jeu fonctionnent correctement.
5. Faire valider par l'enseignant.

1.3 Extension facultative

Pour aller plus loin, vous pouvez ajouter des fonctionnalités supplémentaires, telles que :

- Un système de score pour suivre les points accumulés.
- La possibilité de sauvegarder et de charger une partie.
- Ajout d'une couleur pour chaque tuile lors de l'affichage.