

# Projet de développement d'un Jeu type 2048 like

2048 est un jeu de type puzzle sans adversaire.

Le but du jeu est de faire glisser des tuiles sur une grille, pour combiner les tuiles de mêmes valeurs et en faire la somme et créer ainsi une tuile de valeur supérieure, le but est d'atteindre la création d'une tuile portant le nombre 2048.

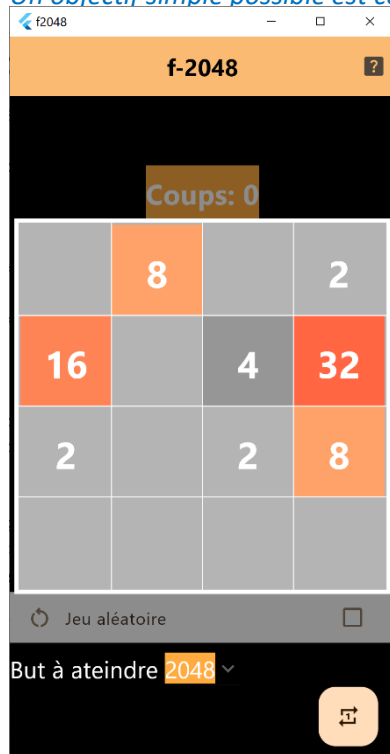
Au départ du jeu, on se retrouve sur un plateau de 16 cases qui comprend déjà deux cases avec des 2. Le principe est assez simple puisqu'il suffit de déplacer les cases avec les touches directionnelles du clavier pour les additionner entre elles. Mais gagner est beaucoup moins évident car il faut bien réfléchir avant de bouger les cases du plateau, au risque de saturer ce dernier, faute de combinaisons disponibles

[https://fr.wikipedia.org/wiki/2048\\_\(jeu\\_vid%C3%A9o\)](https://fr.wikipedia.org/wiki/2048_(jeu_vid%C3%A9o))

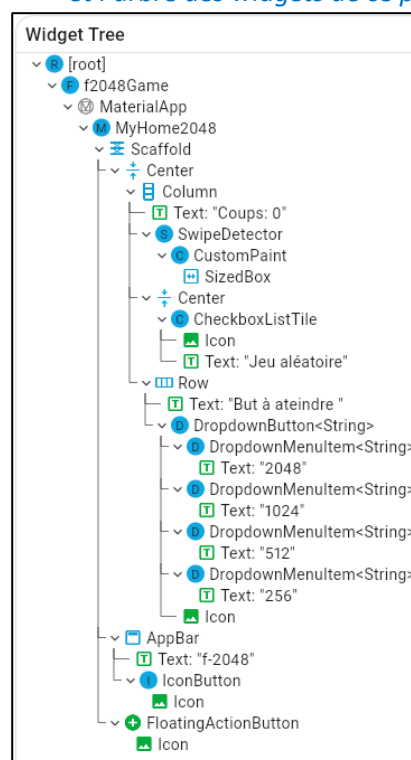
## Projet

Notre projet est ici de développer un jeu de type 2048 avec quelques variantes et options sur les calculs de la grille et de faciliter le jeu permettant à la fois des défis plus abordables pour les joueurs, mais aussi une simplification dans le développement et l'algorithmie de celui-ci.

*Un objectif simple possible est celui-ci*



*et l'arbre des widgets de ce projet*



On présente le jeu avec une « AppBar » simple qui elle-même est constituée du titre du jeux et d'un bouton Action qui permet d'ouvrir une boîte de dialogue « à propos de ».

💡 *Idée : On peut également imaginer mettre un second bouton Action avec les règles du jeu qui sont décrites ci-dessous.*

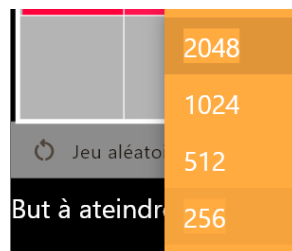
On trouve sur la page centrale :

Un Widget « Text » qui affiche le nombre de coups en cours, (C'est un incrément simple du nombre de gestes),

Un widget « CustomPainter » (que l'on va décortiquer plus tard) permettant d'afficher la grille du jeu en cours,

Une « CheckBoxListTile » qui propose quand elle est cochée de préparer une grille aléatoire avec des chiffres (dans la liste des possibilités du jeu 2048, ces chiffres doivent être inférieurs à la puissance-1 ou -2 de la valeur max décidé),

Un « DropDownButton » et sont « DropDownMenu » permettant de proposer un niveau de difficulté à atteindre (valeurs entre 256 & 2048 par palier de  $2^x$ ).



Et enfin un widget de type FAB ⇔ « FloatingActionButton » permettant de demander un nouveau jeu (aléatoire ou initial selon l'état de la « CheckBox » précédente).

### **Règles :**

La grille comme pour le jeu 2048 classique accepte les valeurs des puissances de 2 (de 1 à 11) soit 2,4,8,16,32,64,128,256,512,1024,2048.

💡 *Idée : allez au-delà de  $2^{11}$  et proposer un mode de jeu plus long et plus complexe en essayant d'atteindre une tuile à 4096 ou 8192 etc...*

Chaque tuile(valeur) dans la grille sera représentée par une couleur différente, ici le choix des couleurs vous revient, classiquement on va du gris à l'orange, mais toutes les palettes sont possibles selon votre envie.

2	4	8	16
32	64	128	256
512	1024	2048	

💡 *Idée : Pourquoi pas ne pas proposer également au joueur de choisir sa palette de couleur sous la forme d'une option.*

La grille doit être sensible aux 4 mouvements (Gestures),

Le balayage (swipe) vers le haut (permet de faire la somme des cases vers le haut et compacter ces chiffres sommés vers le haut,

Le balayage (swipe) vers le bas (permet de faire la somme des cases vers le bas et compacter ces chiffres sommés vers le bas,

Le balayage (swipe) vers la droite (permet de faire la somme des cases vers la droite et compacter ces chiffres sommés vers la droite,

Le balayage (swipe) vers la gauche (permet de faire la somme des cases vers la gauche et compacter ces chiffres sommés vers la gauche,

8		8
16		
16		

Swipe  
Gesture

💡 *Idée : Pourquoi ne pas proposer une option qui permet d'inverser les sommes par rapport aux mouvements (mouvement à gauche, somme vers la droite etc...).*

#### Règles sur les sommes :

On démarre toujours une nouvelle grille avec 2 tuiles, ces tuiles peuvent avoir une valeur de 2 seulement ou des valeurs différentes tiré au hasard (dans les tuiles possibles inférieur à la valeur d'objectif soit  $2^{(\text{objectif}-1)}$ ).

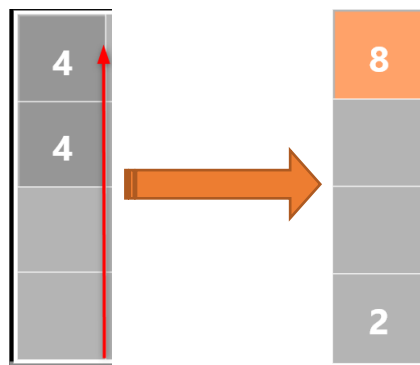
On proposera également ici une possibilité de remplir une grille avec un nombre aléatoire de tuiles qui eux-mêmes auront des valeurs aléatoires, remplir la grille en laissant quelques trous tout de même.

### Mode de calcul

Si le balayage est demandé vers le haut de la grille, la somme des valeurs des 4 colonnes se fera à partir du bas vers le haut.

On réalisera une somme que si et seulement si on a 2 valeurs identiques contiguës

Exemple :



A noter qu'après une somme une nouvelle valeur est insérée dans la grille

Dans le cas de 3 valeurs qui se suivent et un mouvement vers le haut, on effectue la somme des 2 premières valeurs, mais ensuite celle-ci ne match plus avec la 3ème valeur.

Exemple :

4	4	2	2	↑
4	→ 8	ou 2	→ 4	
4	0	2	0	
0	0	0	0	

Autres Exemples :


8	8	8	↑
0	→ 0	→ 4	
2	4	0	
2	0	0	

8	8	8	16	↑
0	→ 0	→ 8	→ 0	
4	8	0	0	
4	0	0	0	


8	8	↑	Ici pas de changements car aucunes sommes possibles.
4	→ 4		
16	16		
2	2		

Dans le cas où la somme des 2 premières valeurs atteint la valeur de la case au-dessus dans ce cas-là sommes résultante est possible également


Exemple avec les différentes étapes déroulées :

8	8	16	
4 →	8 →	0	
4	0	0	
0	0	0	


Autre exemple

8	8	8	16	
4 →	4 →	8 →	0	
2	4	0	0	
2	0	0	0	

Même exemple mais avec un balayage vers le bas


8	8	8	8	
4 →	4 →	0 →	0	
2	0	4	0	
2	4	4	8	

Idem si on réalise une « gesture » de balayage vers la droite




2 2 4 8 → 0 4 4 8 → 0 0 8 8 → 0 0 0 16  
 0 4 4 8 → 0 0 8 8 → 0 0 0 16  
 8 4 2 2 → 8 4 0 4 → 8 0 4 4 → 8 0 0 8

2	2	4	8				16
		4	4	8			16
8	4	2	2				8



Idem si on réalise une « gesture » de balayage vers la gauche



2 2 4 8 → 4 0 4 8 → 4 4 0 8 → 8 0 0 8  
 0 4 4 8 → 0 8 0 8 → 8 0 8 0 → 8 8 0 0 → 16 0 0 0

8 4 2 2 → 8 4 4 0 → 8 8 0 0 → 16 0 0 0

2	2	4	8
	4	4	8
8	4	2	2

8			8
16			
16			

💡 *Idée : Proposer en option différents mode de sommation des chiffres, méthode classique ou méthodes modifié, à vous d'inventer des possibilités.*

#### Etapes de développements :

- 1) Création de la page principale vide
  - a. Mettre en place l'AppBar et le bouton Action « à propos de ».
  - b. Mettre en place le Widget Text pour l'affichage des coups, pour l'instant il affichera un texte statique, à faire évoluer plus tard.
  - c. Mettre en place le bouton FAB (FloatingActionButton) qui permettra de demander un nouveau jeu, mettre une fonction vide pour l'instant qui évoluera bientôt.
- 2) Préparer un Widget de type CustomPainter (CP)
  - a. Préparation, découverte, fonctionnement d'un CP.
    - i. <https://medium.com/flutter-community/a-deep-dive-into-custompaint-in-flutter-47ab44e3f216>
    - ii. [https://api.flutter.dev/flutter/rendering/CustomPainter-class.html?source=post\\_page-----](https://api.flutter.dev/flutter/rendering/CustomPainter-class.html?source=post_page-----)
  - b. Placer ce « CustomPainter » qui n'affichera qu'un carré de couleur couvrant toute la largeur de l'écran dans la page après et en dessous du « Text » statique, ce sera notre zone d'affichage de notre grille de jeu.
- 3) Gestion des « Gestures » sur le « CustomPainter »
  - a. Comment cela fonctionne ?
    - i. <https://docs.flutter.dev/ui/interactivity/gestures>
  - b. Utilisation du package « flutter\_swipe\_detector.dart »
    - i. ([https://pub.dev/packages/flutter\\_swipe\\_detector](https://pub.dev/packages/flutter_swipe_detector)) a partir de l'exemple du package, adaptez ces lignes de codes afin qu'elles s'intègrent dans votre application.
  - c. Mise en place des « gestures » et capture des directions.
    - i. Celle-ci doit utiliser comme enfant le CP que l'on a créé précédemment afin de pouvoir donner à ce CP un pouvoir de détection de balayage et de réagir au sens du balayage.

- 4) Gestion de la grille des valeurs (quel modèle ? comment faire ?)
  - a. Mise en place d'une liste d'entier List<int> (tableau de 16 cases qui pourront prendre les valeurs 0 ⇔ case vide, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048)
  - b. Gestion des coordonnées
 

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Ligne0 → 0 à 3, Ligne1 → 4 à 7, Ligne2 → 8 à 11, Ligne3 → 12 à 15

Col0 → 0,4,8,12 : Col1 → 1,5,9,13 : Col2 → 2,6,10,14 : Col3 → 3,7,11,15

Plus généralement :

Ligne X(0,3) :  $(X*4) + 0, (X*4) + 1, (X*4) + 2, (X*4) + 3$

Col Y(0,3) :  $Y + (0*4), Y + (1*4), Y + (2*4), Y + (3*4)$

Mise en place du constructeur du CP (pour passer en paramètre la List<int> qui nous permettra de représenter visuellement cette liste sous la forme d'une grille 4x4.

- c. Dessin de la grille sur le « CustomPainter ».
    - i. Une couleur de fond pour effacer l'écran
    - ii. Dessiner le cadre du contour d'une épaisseur de 4 ou 6 pixels.
  - d. Affichage des nombres sur le CP et des messages au joueur.
    - i. Dessiner les 16 valeurs (si 0 alors case vide)
      1. Faire un test avec une grille de DEBUG
      2. Ajuster vos couleurs de tuiles.
      3. Centrer le texte de chaque tuile dans la case.
    - ii. Dessiner les lignes séparatrice (1 ou 2 pixels d'épaisseur).
    - iii. Dessiner le Texte du message si non vide (Win ou End).
- 5) Mise en place de l'algorithme de calcul de la grille.
- 6) Mise en place de la capture des événements (fonctions de test et conditions d'arrêts du jeu).
  - a. Gagné !
  - b. Perdu, plus de mouvements !
  - c. Le jeu continue...
- 7) Mise en place du nouveau jeu
  - a. Nouveau jeu normal (vider la grille, placer 2 valeurs dans 2 cases au hasard).
  - b. Nouveau jeu avec grille aléatoire, « CheckBoxListTile » (plusieurs valeurs dans plusieurs case au hasard).
- 8) Mise en place du Goal à atteindre (niveau de jeu)
  - a. Mise en place du « DropdowButton » et son menu « DropDownMenu »

- b. Modifier la fin de jeu gagnant afin de prendre en compte la valeur objectif de la variable Goal.
- c. Mise en place de l’affichage du nombre de coups.
  - i. Positionner une variable globale qui à chaque coups va s’incrémenter et sera afficher dans notre WidGet « Text() »

💡 Idée : Permettre une sauvegarde et relecture des performance du joueurs, nombre coups minimum pour un objectif dédié.

💡 Idée : placer les paramètres de jeux dans une page dédiée au paramétrage.

💡 Idée : une fois l’objectif atteint, permettre au joueur de continuer le jeux afin de sommer les cases de sa grille (ce que propose le jeu normal).

💡 Idée : Améliorer le design de la grille pour quelle ait un aspect plus moderne,

💡 Idée : proposer un « Splash Screen » au lancement du jeu.

💡 Idée : proposer une grille plus large de 8x8 par exemple en option.



*Vous pourriez poursuivre ce TP en tant que projet de fin d’étude et y apporter un maximum d’améliorations, selon les idées proposées tout au long du TP, mais également vos propre idées d’améliorations ou de dérivation de ce jeu.*

Remise du TP au plus tard le Lundi 09 décembre 2024 par messagerie du lien GitHub, uniquement les codes sources accompagnés d’un document comportant les noms des personnes ayant participées au groupe de travail (Nom + Prénom + Num étudiant)

Lien à envoyer aux adresses obligatoirement (nb :les deux adresses) ...

[herve.tondeur@uphf.fr](mailto:herve.tondeur@uphf.fr)

[tondeur.herve@yahoo.fr](mailto:tondeur.herve@yahoo.fr)

A noter :

**Vous pouvez supprimer les dossiers**

.\build

.\dart tool\flutter build

Et tout dossier qui vous semble superflu de transmettre dans votre GITHUB