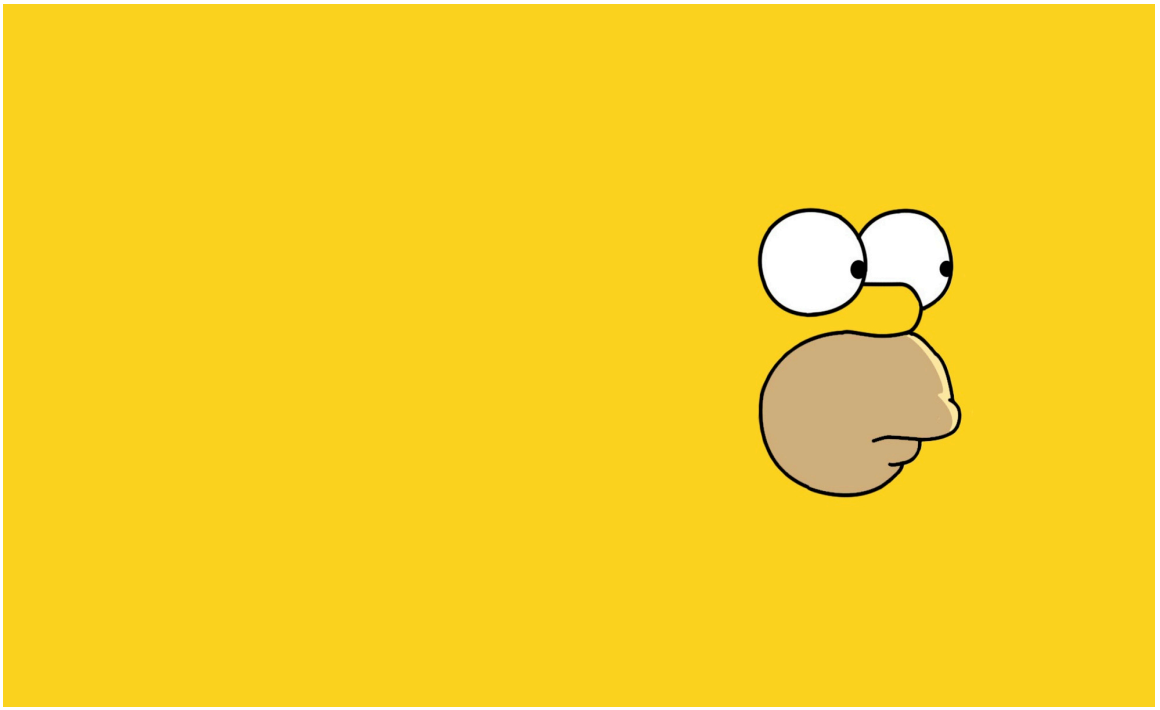
2048

Rapport de l'application mobile du 2048



GUARIM Raphäel & GUILLIER Camille

04/09/2025
FISA 5

INTRODUCTION

Ce projet a été l'occasion d'aborder Flutter sous un angle plus créatif. Après plusieurs travaux techniques, il s'agissait ici de concevoir une application plus libre, en revisitant un concept connu : le 2048. Le but n'était pas seulement de reproduire le jeu, mais d'en proposer une version personnelle en intégrant de nouvelles idées de design et de gameplay.

Ce format nous a permis de revoir les bases du développement Flutter à travers un jeu simple, tout en expérimentant sur l'organisation du code, la gestion d'état et l'intégration d'éléments visuels. Le projet mettait davantage l'accent sur la créativité et la cohérence de l'ensemble que sur la complexité technique.

Fonctionnalité spécifique application

Notre version du 2048 se distingue principalement par sa direction artistique et par l'ajout d'éléments de gameplay.
Nous avons choisi un thème "Homer Simpson" : chaque tuile représente une étape de la vie d'un Simpson, qui évolue au fil des fusions, depuis le spermatozoïde jusqu'au vieux Homer. Ce choix apporte une dimension humoristique et visuelle, en remplaçant les chiffres habituels par des images progressives.

En plus du thème graphique, nous avons intégré des tuiles spéciales qui modifient le déroulement du jeu :

- la **tombe**, qui agit comme un malus en supprimant la tuile voisine du plateau ;
- le **bar**, qui agit comme un bonus en doublant la tuile voisine.

Ces éléments rendent les parties plus dynamiques et imprévisibles, en introduisant une part de stratégie et de hasard supplémentaire. L'apparition aléatoire de ces tuiles oblige le joueur à adapter sa façon de jouer et rend chaque partie unique.

Enfin, le jeu a été pensé pour être responsive, avec une interface fluide adaptée aussi bien aux smartphones qu'aux tablettes. L'organisation du code (séparation entre logique de jeu, gestion des styles et interface) permet une bonne lisibilité et facilite les futures extensions.

Aperçu visuel

Explication technique

Structure principale & Logique de jeu

Les deux classes principales _GameState & Tile vont respectivement contenir l'ensemble des informations du jeu ainsi que des widgets visuels tell que l'écran, le texte pour le score ainsi que le bouton pour recommencer, La seconde classe va permettre d'avoir les informations d'une tuile(case) pour savoir sa valeur, l'image associé et donc de pouvoir mettre un design personnalisé de cette tuile.

StatefulWidget est utilisé pour pouvoir modifier dynamiquement la grille et le score à chaque mouvement. La méthode setState() permet de redessiner l'interface à chaque action (ajoute de tuile, fusion et le reset).

GridView.builder permet de créer la grille en générant les 16 cases en fonction de la matrice dans le code. Parmis les 16 emplacement dans la grille, il peut y avoir plusieurs nombres :

- 2 4 8 16 32 64 128 256 512 1024 et 2048, il ont tous une couleur et une image qui corresponde à leur nombre, récupérer grâce à la fonction _tileStyle
- 1 et 3 qui sont les bonus / malus
- 0 qui est un emplacement libre

GestureDetector détecte les gestes du joueur (haut, bas, droite et gauche). Une fois qu'on sait que l'utilisateur a fait un glissement, on regarde si c'est un glissement assez important, et dans quelle direction il est effectué, en fonction de ça on va appeler des fonctions différentes.

interface dynamique

Container & Padding organisent et espacent les éléments pour un meilleur rendu. Grâce au widget **AnimatedContainer**, on va pouvoir avoir une rapide animation sur le changement des couleurs.

AspectRatio va permettre de s'assurer que la grille reste carrée, quelle que soit la taille de l'écran