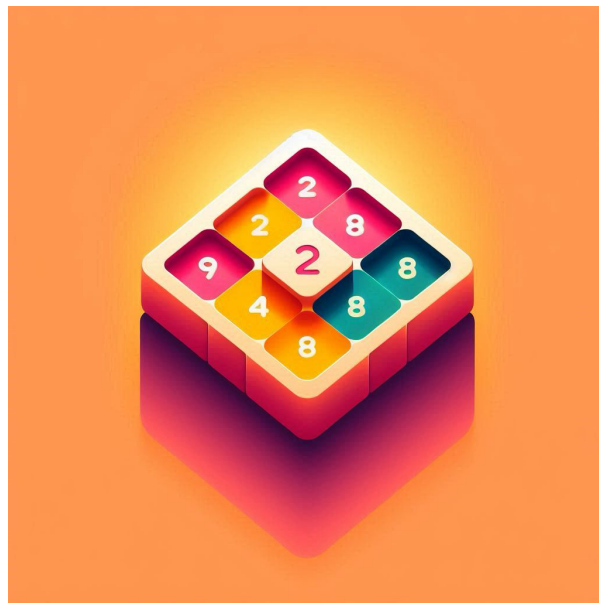


Rapport du projet : 2048 Like



Réalisé et présenté par :

Nom et Prénom	Numéro étudiant
AGOKOLI Noudehouenou Don Arias	22403861
MALBET Corentin	22400059
THERET Tom	22403926

Introduction	3
Structure du project	3
pubspec.yaml :	3
main.dart :	3
game_controller.dart :	3
app_theme.dart.....	4
Fonctionnalités principals.....	4
Logique du jeu :	4
Expérience utilisateur:	4
Gestion d'état:	4
Capture d'écran :	5
GitHub Actions.....	6
Ficher README.md	6
Axes d'amélioration	6
Conclusion	7

Introduction

Ce projet est une implémentation du jeu **2048**, développé dans le cadre du cours de développement mobile avec le Framework Flutter. Le projet vise à enseigner les principes de développement d'applications multiplateformes, de gestion d'état et l'intégration de fonctionnalités avancées comme des animations, des sons, et des interactions utilisateur.

Vous pouvez accéder au code source du projet sur GitHub en suivant ce lien : <https://github.com/dongorias/Tp2048Like>.

Structure du project

`pubspec.yaml` :

- Définit les dépendances principales:
 - **confetti** pour des animations de célébration.
 - **provider** pour la gestion de l'état.
 - **audioplayers** pour la gestion des sons
 - **text animator** pour l'animation du texte
 - **flutter svg** pour afficher des icônes en fichier svg
- Inclut des fichiers multimédias dans le dossier `assets/` (svgs, sounds, fonts, images, icônes).
- Configuration des icônes via **icons_launcher**.

`main.dart` :

- Initialise l'application avec **MultiProvider** pour la gestion de l'état.
- Définit les routes pour naviguer entre les écrans :
 - `/intro`: Écran d'introduction.
 - `/game` : Écran principal du jeu.
 - `/about` : Écran à propos.
- Configure le thème de l'application.

`game_controller.dart` :

- Contrôle la logique principale du jeu :
 - Gestion du plateau (4x4), des mouvements (gauche, droite, haut, bas) et des règles.
 - Détection de la fin de partie (victoire/défaite).
 - Gestion des animations, sons et interactions utilisateur.
- Implémente la logique de fusion des tuiles et de génération aléatoire.
- Intègre un mécanisme de timer

- Intégration de confettis pour des animation visuels quand l'objectif atteints.

app_theme.dart

Le fichier app_theme.dart définit les styles et thèmes personnalisés de l'application. Il contient une extension sur BuildContext pour accéder facilement aux éléments du thème, tels que les couleurs, les polices et les styles de texte. Ce fichier centralise les configurations de la palette de couleurs, des tailles de police, et des styles typographiques, comme les titres, les boutons, le corps du texte et les légendes, avec une intégration d'une police personnalisée 'Electrolize'.

Fonctionnalités principales

Logique du jeu :

- Implémente les règles de base du 2048 :
 - Fusion des tuiles similaires.
 - Génération de nouvelles tuiles après chaque mouvement.
 - Détection des conditions de victoire et de défaite.

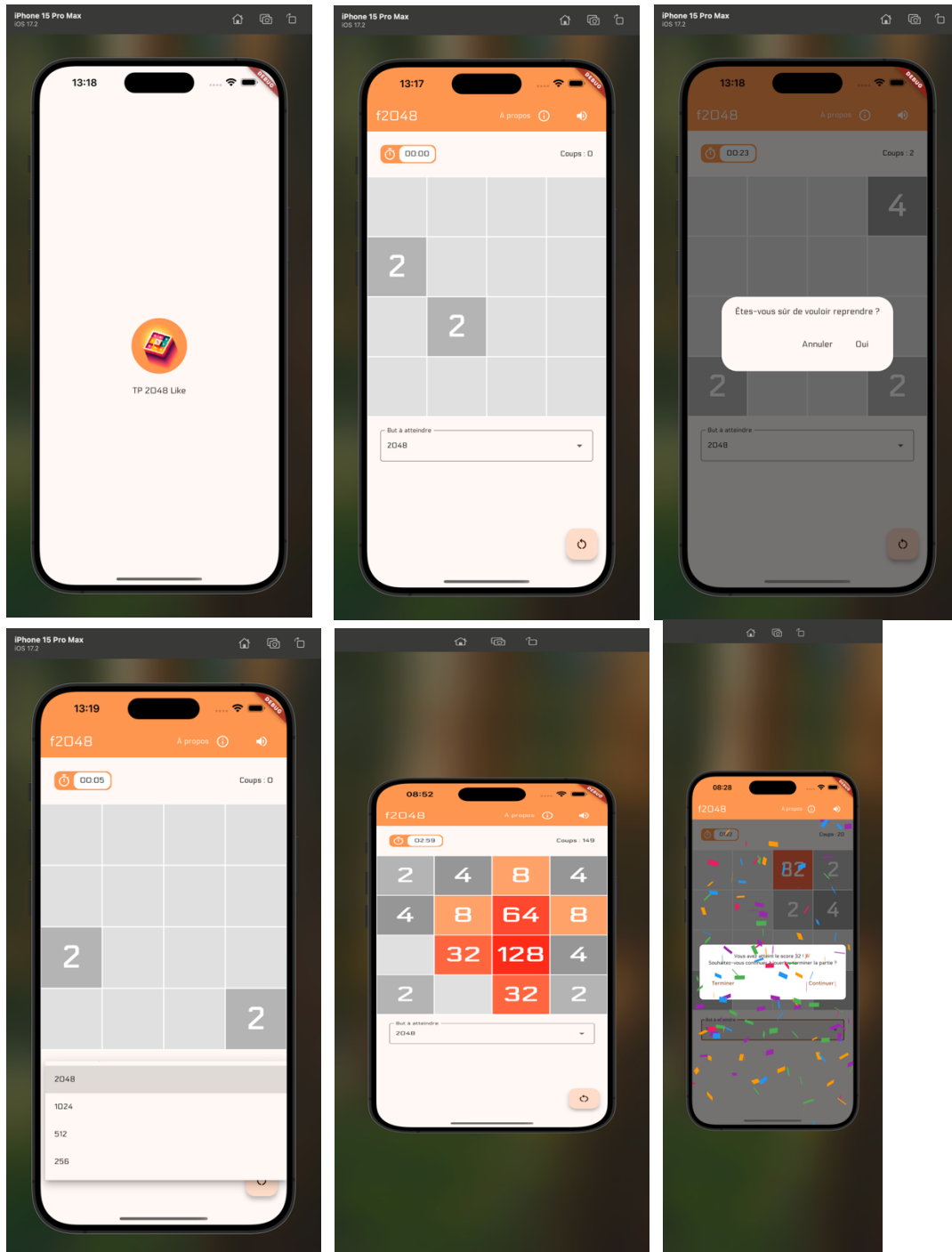
Expérience utilisateur:

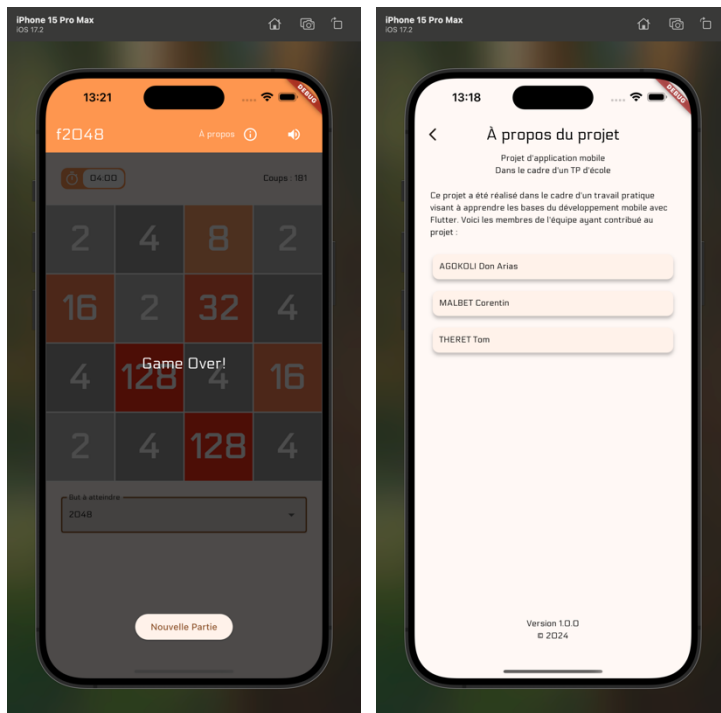
- Utilise des animations visuelles pour améliorer l'expérience (confettis pour une victoire).
- Ajoute des effets sonores contextuels aux actions de jeu, tels que victoire, défaite ou glissement, en fonction du type d'action.

Gestion d'état:

- Utilisation de **Provider** pour réagir aux changements dans le modèle de données et mettre à jour l'interface en conséquence.

Capture d'écran :





GitHub Actions

Un workflow GitHub Actions est configuré pour générer automatiquement les APK de release.

Ficher README.md

Le fichier README accompagnant le projet fournit une introduction détaillée, des captures d'écran illustrant l'application, ainsi qu'une liste exhaustive des fonctionnalités, notamment les effets sonores, les animations et les modes de jeu personnalisables. Il décrit également les technologies et plugins utilisés, les étapes d'installation, les objectifs pédagogiques du projet, les membres de l'équipe, les informations sur la licence et les exigences système. De plus, il inclut des détails sur le déploiement multiplateforme et l'intégration d'un workflow pour la génération automatique d'APK.

Axes d'amélioration

- Ajouter une option pour sauvegarder et restaurer les parties en cours via **shared_preferences**.
- Intégrer un système de score en ligne pour comparer les performances des joueurs.

- Option pour activer ou désactiver la logique pour générer aléatoirement des tuiles avec une probabilité ajustée (90% pour 2, 10% pour 4).

Conclusion

Ce projet démontre efficacement les capacités de Flutter pour développer un jeu interactif, combinant logique de jeu complexe et intégrations multimédias avancées. Il constitue un excellent outil pédagogique pour explorer des concepts clés comme la gestion d'état et les animations.