

FONCTIONNALITÉS DU SONGHOGAME

Le jeu **SONGHOGAME** est développé en langage Flutter (Dart) Le but du jeu est de s'emparer d'un maximum de graines. Le joueur ayant le plus de graines à la fin de la partie l'emporte.

Le jeu Songho comporte les règles suivantes :

- La distribution est l'opération qui consiste à prendre la totalité des billes présentes dans un trou de son camp et à les disposer une à une dans les cases qui le suivent dans l'ordre inverse des aiguilles d'une montre (on peut donc distribuer aussi dans les cases de son adversaire).
- Une prise est l'opération qui consiste à ramasser la totalité des graines d'un trou dans le camp de l'adversaire car les prises se font uniquement chez son adversaire.
- Si la dernière graine distribuée tombe dans un trou de l'adversaire comportant 2 ou 3 graines (avec la distribution effectuée), le joueur capture les 2 ou 3 graines résultantes.
- Capture multiple : lorsqu'un joueur s'empare de deux ou trois graines, si la case précédente contient également deux ou trois graines, elles sont capturées aussi, et ainsi de suite.
- Bouclage : si le nombre de graines prises dans le trou de départ est supérieur à 11, cela fait que l'on va boucler un tour : auquel cas, à chaque passage, la case de départ est sautée et donc laissé vider pour cette distribution laissée vide. Un trou contenant assez de graines pour faire une boucle complète s'appelle un **Krou**.
- Donner à manger : on n'a pas le droit d'affamer l'adversaire. De même, un joueur n'a pas le droit de jouer un coup qui prenne toutes les graines du camp de l'adversaire.
- Si le nombre de pierres pris par l'un des joueurs est supérieur à 35, il gagne
- Si le nombre de pierres des deux joueurs, en excluant les prises, est inférieur à 10, le gagnant est celui qui a la somme de ses prises et du nombre des pierres de son côté, supérieur à 35.

MODE ONEPLAYER

Dans ce mode, l'utilisateur est amené à faire une partie de jeu contre l'ordinateur. Il a été implémenté sans base de données, donc les données de jeu sont sauvegardées en local sur le mobile.

Plusieurs algorithmes sont implémentés pour ce mode à savoir :

- Une widget **buildCell()** : qui prend en paramètre le nombre de pions et les couleurs des krou, permet de générer les Krou pour les pions.
- A l'intérieur de ce widget, on a la fonction **onTapBtn()** qui s'exécute lorsqu'on touche à un cell généré par le widget buildCell().
- A l'intérieur de cette fonction onTapBtn(), on a principalement la fonction **_distributePawns()** qui fait la distribution des pions dans les différents

krou, cette fonction prends en paramètre les index des krou et effectue l'incrémentation des pions de chaque krou.

- A l'intérieur de la fonction `_distributePawns()`, on a plusieurs algorithmes parmi lesquels : un **while** qui vérifie que le nombre de pions de chaque krou est supérieur à 0, et chaque fois que chaque krou contient un nombre de pions inférieur à 0, il récupère ces pierres et fait la distributions dans les autres krou.
- On a aussi la fonction **`_computerDistributePawns()`** qui permet à l'ordinateur de faire sa distribution dans les krou.
- On a aussi des boucles qui vérifie à chaque fois les scores des joueurs et affiche le résultat à l'écran selon que le joueur 1 ou l'ordinateur ait gagné ou perdu.

MODE TWOPLAYER

Dans ce mode, l'utilisateur est amené à faire une partie de jeu contre un autre joueur sur le même appareil. Il a été implémenté sans base de données, donc les données de jeu sont sauvegardées en local sur le mobile.

Plusieurs algorithmes sont implémentés pour ce mode à savoir :

- Une widget **`buildCell()`** : qui prend en paramètre le nombre de pions et les couleurs des krou, permet de générer les Krou pour les pions.
- A l'intérieur de ce widget, on a la fonction **`onTapBtn()`** qui s'exécute lorsqu'on touche à un cell généré par le widget buildCell().
- A l'intérieur de cette fonction `onTapBtn()`, on a principalement la fonction **`_distributePawns()`** qui fait la distribution des pions dans les différents krou, cette fonction prends en paramètre les index des krou et effectue l'incrémentation des pions de chaque krou.
- A l'intérieur de la fonction `_distributePawns()`, on a plusieurs algorithmes parmi lesquels : un **while** qui vérifie que le nombre de pions de chaque krou est supérieur à 0, et chaque fois que chaque krou contient un nombre de pions inférieur à 0, il récupère ces pierres et fait la distributions dans les autres krou.
- On a aussi des boucles qui vérifie à chaque fois les scores des joueurs et affiche le résultat à l'écran selon que le joueur 1 ou le joueur 2 ait gagné ou perdu.

MODE ONLINE

Ce mode est développé suivant le principe du jeu **LUKOKING** où pour lancer une partie en ligne, on doit au se sert d'un code qu'on partage aux autres joueurs et ils s'en servent pour intégrer la partie.

Pour ce qui est du songhgame, c'est le même principe, nous avons implémenter les fonctions suivantes pour la réalisation du jeu en mode online :

- Fonction **generateHashCode()** qui génère un code (hexadécimal) à 4 valeurs
- La fonction **saveHashCode()** qui effectue la sauvegarde en base de données, du code généré précédemment.

Tout ceci se fait dans cet onglet :



En cliquant sur ce bouton, un code est généré et affiché à l'écran. Ensuite l'initiateur de la partie le partage aux autres joueurs à travers une quelconque plateforme de communication ; par la suite, la personne qui reçoit le code, doit le saisir dans la partie de la page réservée à l'accèsion d'une partie.



- Dans le champ de saisi, vous entrez le code qui vous a été envoyé et lorsque vous finissez la saisie, la fonction **getValueFromTextField()** s'exécute.

En cliquant sur le bouton noir, d'autres algorithmes sont exécutés en arrière plan à savoir :

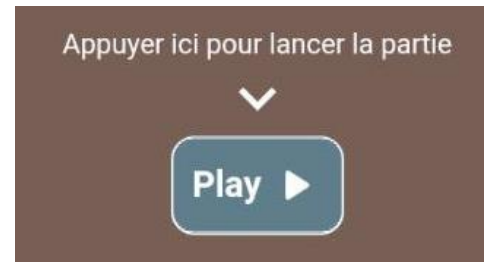
- Une fonction **onPressedButton()** qui ne prend aucune donnée en paramètre mais elle exécute d'autres fonctions qui y sont incluses à savoir :

```
result = await saveUidIfHashCodeExists(inputHashCode, user.uid);
```

Cette ligne permet donc de faire la recherche de code saisi dans la base de données du jeu. Lorsque la recherche est positive, il y a une série d'actions qui sont exécutées à savoir :

- L'enregistrement de l'id du joueur qui souhaite rejoindre la partie dans la base de données.
- Il récupère aussi le nom du joueur et l'affiche dans la fenêtre appropriée pour le nouveau joueur.

Après avoir fait tout ceci, il y a le bouton jouer qui permet de lancer la partie de jeu lorsqu'un deuxième joueur remplit toutes les formalités, pour plus de sécurité, nous avons inclus une fonction derrière ce bouton qui permet effectivement de vérifier qu'effectivement un deuxième personne à rejoint la partie et cela se fait en vérifiant que son identifiant se trouve dans la base de données, qu'un code de partie a été généré par le premier joueur et sauvegardé, et enfin que lui aussi (le premier joueur), son id a été enregistré dans la base de données. Sur la base de tout ceci, le jeu va démarrer comme ceci :



Pour que tout ceci se fasse, il doit au préalable avoir un compte car le mode online se joue avec des comptes. Donc nous avons implémenter des interfaces d'inscription et de connexion dans le jeu.

- **Inscription** : à l'intérieur de notre projet, nous avons un fichier *auth_controller.dart* qui contient toutes les fonctions de connexion et/ou inscription. Pour ce cas, nous avons implémenter trois fonctions à savoir :
 - **pickImage()** : qui permet à l'utilisateur de choisir une image de profil dans sa galerie.
 - **_uploadToStorage()** : qui permet à l'utilisateur d'enregistrer son image dans la base de données.
 - **register()** : qui permet de récupérer les différentes informations d'inscription et de les sauvegarder en ligne.
- **Connexion** : Pour la connexion, nous avons une fonction **login()** qui prend en paramètre l'adresse e-mail et le mot-de-passe, cette fonction récupère ces deux données dans des champs et procède à une vérification avec ce qui existe dans la base de données. Au cas où les informations correspondent, le joueur est connecté et peut désormais jouer en ligne.