

Etat du Sprint 2 Réalisations & Objectifs

Objectifs du Sprint 2

- Audio réel : MediaPlayer + API iTunes, lecture fluide, fallback playlist locale.
 - UI complète : MainMenu, GameView, LeaderboardView, SettingsView, boutons supplémentaires (“Duel”, “Classement”, “Paramètres”, “Charger playlist”).
 - Logique de jeu avancée : scoring détaillé, gestion joueurs, modes Solo/Duel, indices, sélection de genres.
 - Persistance : sauvegarde scores et paramètres JSON, chargement playlists.
 - Tests et validation : tests unitaires et manuels, Javadoc complète.
-

Réalisations

Répartition des 100 points

- **Iskander** : 35 points — AudioService complet, base solide pour l'intégration.
 - **Léo** : 40 points — Persistance JSON complète, structure stable et réutilisable.
 - **Achraf** : 25 points — UI partiellement réalisée, logique de jeu basique implémentée.
 - Audio : AudioService fonctionnel (Iskander).
 - Persistance : Services JSON complets (Léo).
 - UI :
 - MainMenu terminé en partie, manque de boutons (Achraf).
 - GameView, LeaderboardView et SettingsView partiellement développés (Achraf).
 - Logique de jeu :
 - Scoring basique et gestion de joueurs.
 - Fonctionnalités avancées (Solo/Duel, indices, genres) non intégrées (Achraf).
 - Tests : Tests présents, Javadoc complétée, mais il manque certains tests concernant l'UI et les interactions.
-

Écarts

- UI non finalisée.
- Modes Solo/Duel, indices et genres manquants.

- Intégration audio/persistance dans l'UI en attente (dépend de vues à compléter).
 - Tests et documentation incomplets.
-

Conclusion Sprint 2

Environ 50 % réalisé.

Les services audio et persistance sont solides, mais l'UI et la logique de jeu doivent être finalisées pour le Sprint 3.

Plan pour le Sprint 3

Objectifs

- Finaliser l'UI complète : GameView, LeaderboardView, SettingsView, intégration des boutons et navigation.
 - Implémenter la logique de jeu avancée : modes Solo/Duel, indices, sélection de genres, timer, saisie des réponses.
 - Intégrer audio et persistance dans l'UI : lecture réelle, sauvegarde automatique, chargement playlists.
 - Réaliser des tests complets : unité + manuel, Javadoc complète.
 - Documentation finale : README, manuel utilisateur, diagrammes mis à jour.
-

Répartition des tâches détaillée

Iskander

- Finaliser l'intégration audio dans GameController et l'UI (continuation du Sprint 2).
- Implémenter la sélection de genres dans l'API iTunes.
- Développement UI : GameView (timer, saisie réponses), intégration des boutons dans MainMenu.
- Tests unitaires pour AudioService et intégration.

Léo

- Finaliser la persistance : leaderboard, historique joueurs, chargement playlists JSON (continuation Sprint 2).

- Implémenter le scoring détaillé (bonus rapidité).
- Développer l'UI : LeaderboardView, SettingsView (paramètres audio, indices).
- Tests unitaires pour ScoreService et PersistenceService.
- Intégration persistance dans l'UI : sauvegarde automatique, chargement playlists.

Achraf

- Implémentation des modes Solo/Duel et indices (logique de jeu avancée).
 - Tests manuels de l'UI et validation des interactions.
 - Tests unitaires supplémentaires pour la logique de jeu.
 - Rédaction partielle de la documentation (manuel utilisateur, diagrammes).
 - Polishing et optimisation UI (améliorations non critiques).
-

Livrable attendu

Application BlindTest complète et jouable :

Modes Solo/Duel, audio réel via iTunes API, scoring persistant, UI fluide, tests couvrant la logique métier, documentation complète.

Projet compilable et exécutable sans erreurs.

Risques

- Compatibilité JavaFX selon les OS.
 - API iTunes inaccessible : fallback déjà prévu.
 - Charge équilibrée via revues hebdomadaires.
-

Lien GitHub

<https://github.com/illeska/blindtest>