

## État du Sprint 1 : Réalisations vs Objectifs

### Objectifs du Sprint 1 :

- Mettre en place le projet et l'environnement : dépôt Git, Gradle, structure des packages (app, library), dossiers (src/, doc/, data/, test/) et fichiers de base (README, .gitignore).
- Développer un prototype fonctionnel minimal : écran d'accueil, classes de base (Player, Track), GameController avec startGame() et nextRound(), Round avec simulation audio, Playlist et PlaylistService pour playlists JSON, tests de communication entre composants.
- Livrable : projet compilable, prototype minimal, dépôt Git avec commits réguliers.

### Réalisations :

- **Mise en place du projet** : structure modulaire Gradle, packages organisés, dossiers créés, README et .gitignore présents.
- **Classes de base** : Player et Track implémentés, Round avec mock audio, Playlist et PlaylistService fonctionnels, GameController avec gestion de rounds.
- **UI de base** : MainMenu avec boutons "Jouer" et "Quitter".
- **Tests** : unitaires pour Player et GameController.
- **Collaboration** : commits réguliers, activité suivie.

### Écarts :

- Prototype minimal, pas de saisie de réponses ni scoring réel.
- PlaylistService non intégré dans l'UI.
- Tests partiels, documentation incomplète.
- Dépendances JUnit manquantes causant des erreurs de compilation.

**Conclusion Sprint 1** : ~70–80 % réalisé. Prototype fonctionnel, mais intégration UI et tests complets à compléter pour Sprint 2.

---

## Plan pour Sprint 2

### Objectifs :

- **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.

### **Répartition des tâches détaillée :**

- **Iskander** :
  - Intégration API iTunes et MediaPlayer.
  - Développement d'AudioService avec gestion des erreurs et fallback.
  - Implémentation du playback réel dans GameController.
  - Tests unitaires pour AudioService.
- **Achraf** :
  - Développement complet de l'UI : GameView, LeaderboardView, SettingsView.
  - Intégration des boutons supplémentaires dans MainMenu.
  - Implémentation du timer et de la logique de saisie des réponses.
  - Tests manuels de l'UI et validation des interactions.
- **Léo** :
  - Gestion de la persistance : sauvegarde/chargement JSON pour scores, settings et playlists.
  - Développement du Leaderboard et historique des joueurs.
  - Tests unitaires pour persistance et intégration scoring.
  - Rédaction de la documentation et Javadoc complète.

**Livrable attendu** : application jouable avec modes Solo/Duel, audio réel, scoring, sauvegarde et tests.

**Risques** : API inaccessible (fallback prévu), compatibilité JavaFX sur différents OS, charge équilibrée via revues hebdomadaires.

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