Documentation du Projet d'Application Musicale

Ce document répertorie toutes les fonctionnalités de l'application musicale et décrit les tests qui ont été effectués pour assurer son bon fonctionnement.

Table des matières

Architecture du projet

L'application suit le modèle d'architecture Modèle-Vue-Contrôleur (MVC) et est organisée en modules Maven :

- **Main** : Point d'entrée de l'application
- **Model** : Logique métier et modèles de données
- **View** : Composants de l'interface utilisateur
- **Controller** : Contrôleurs qui connectent le modèle et la vue
- **Share** : Ressources partagées et utilitaires

Fonctionnalités

Instruments de musique

1. Instruments MIDI

L'application utilise la classe abstraite MidiInstrument pour implémenter des instruments basés sur MIDI :

- **Piano** (MIDI instrument 0)

- **Xylophone** (MIDI instrument 13)
- **Acoustic Guitar** (MIDI instrument 24)
- **Electric Bass** (MIDI instrument 33)
- **Wood Instrument** (MIDI instrument 116)

Ces instruments utilisent l'API Java Sound pour produire des sons MIDI avec différents timbres.

2. Batterie (DrumKit)

La classe DrumKit implémente un instrument de batterie qui utilise des échantillons audio WAV pour produire des sons de batterie :

- Grosse caisse (notes 35, 36)
- Caisse claire (notes 38, 40)
- Charleston fermé (note 42)
- Charleston ouvert (note 46)
- Cymbale crash (note 49)
- Cymbale ride (note 51)

3. Instruments à forme d'onde

La classe abstraite SinusAudio permet de créer des instruments basés sur la génération directe de formes d'onde :

- **Video Game** : Utilise des ondes carrées pour produire des sons similaires à ceux des jeux vidéo rétro

Interface utilisateur

1. Fenêtre principale

La classe Frame fournit la fenêtre principale de l'application avec :

- Un en-tête avec des boutons de navigation
- Un panneau central pour afficher les instruments
- Un panneau d'enregistrement (lorsqu'un instrument est affiché)

2. Panneaux d'instruments

Chaque instrument dispose d'un panneau dédié :

- **PianoPanel** : Interface pour jouer du piano
- **XylophonePanel** : Interface pour jouer du xylophone
- **BitPanel** : Interface pour l'instrument Video Game
- **WoodPanel** : Interface pour les instruments en bois
- **DrumPanel** : Interface pour jouer de la batterie

3. Contrôles

L'application permet de :

- Changer d'instrument
- Ajuster l'octave
- Ajuster la vélocité (volume/intensité) des notes

Enregistrement et lecture

1. Enregistrement

La classe Record permet d'enregistrer les performances musicales :

- Enregistre les notes jouées avec leur durée
- Enregistre les pauses entre les notes
- Sauvegarde les enregistrements au format JSON

2. Lecture

La classe OuvrirPartition permet de :

- Ouvrir des fichiers JSON contenant des enregistrements
- Lire les enregistrements
- Visualiser les notes jouées

Procédures de test

Tests des instruments

Test du Piano et autres instruments MIDI

1. **Test de base** : Vérifier que chaque note produit un son lorsqu'elle est jouée
    - Jouer chaque note du clavier virtuel
    - Vérifier que le son correspond à la note attendue
    - Vérifier que le son s'arrête lorsque la note est relâchée

2. **Test d'octave** : Vérifier que le changement d'octave fonctionne correctement
    - Jouer une note, puis changer d'octave et jouer la même note
    - Vérifier que la hauteur du son change en conséquence

3. **Test de vélocité** : Vérifier que le changement de vélocité affecte le volume
    - Jouer une note avec différentes valeurs de vélocité
    - Vérifier que le volume change en conséquence

Test de la Batterie (DrumKit)

1. **Test des sons** : Vérifier que chaque élément de batterie produit le son correct
    - Jouer chaque élément de batterie (grosse caisse, caisse claire, etc.)
    - Vérifier que le son correspond à l'élément attendu

2. **Test de lecture simultanée** : Vérifier que plusieurs sons peuvent être joués simultanément
    - Jouer rapidement plusieurs éléments de batterie
    - Vérifier que tous les sons sont joués sans interruption

Test de l'instrument Video Game

1. **Test des ondes carrées** : Vérifier que les sons produits ont la forme d'onde attendue
    - Jouer différentes notes
    - Vérifier que les sons ont la qualité "rétro" caractéristique des ondes carrées

Tests de l'interface utilisateur

Test de la fenêtre principale

1. **Test de navigation** : Vérifier que les boutons de navigation fonctionnent correctement
    - Cliquer sur chaque bouton de l'en-tête
    - Vérifier que le contenu de la fenêtre change en conséquence

2. **Test de changement d'instrument** : Vérifier que le changement d'instrument fonctionne correctement
    - Sélectionner différents instruments
    - Vérifier que le panneau d'instrument correspondant s'affiche
    - Vérifier que l'instrument sélectionné produit les sons attendus

Test des panneaux d'instruments

1. **Test d'interaction** : Vérifier que les interactions avec les panneaux d'instruments fonctionnent correctement
    - Cliquer sur les éléments interactifs (touches, pads, etc.)
    - Vérifier que les sons correspondants sont produits

2. **Test de nettoyage des ressources** : Vérifier que les ressources sont correctement libérées lors du changement de panneau
    - Changer d'instrument plusieurs fois
    - Vérifier qu'il n'y a pas de fuites de mémoire ou de ressources

Tests d'enregistrement et de lecture

Test d'enregistrement

1. **Test de base** : Vérifier que l'enregistrement fonctionne correctement
    - Commencer l'enregistrement
    - Jouer une séquence de notes
    - Arrêter l'enregistrement
    - Vérifier que le fichier JSON est créé et contient les notes jouées

2. **Test de durée** : Vérifier que les durées des notes et des pauses sont correctement enregistrées
    - Enregistrer une séquence avec des notes de différentes durées et des pauses

- Vérifier que les durées dans le fichier JSON correspondent aux durées jouées

Test de lecture

1. **Test de base** : Vérifier que la lecture fonctionne correctement
    - Ouvrir un fichier JSON enregistré
    - Lancer la lecture
    - Vérifier que les notes sont jouées dans le bon ordre et avec les bonnes durées

2. **Test d'arrêt** : Vérifier que l'arrêt de la lecture fonctionne correctement
    - Lancer la lecture d'un enregistrement
    - Arrêter la lecture avant la fin
    - Vérifier que la lecture s'arrête immédiatement

Tests d'intégration

1. **Test de flux complet** : Vérifier que le flux complet de l'application fonctionne correctement
    - Lancer l'application
    - Sélectionner un instrument
    - Jouer quelques notes
    - Enregistrer la performance
    - Sauvegarder l'enregistrement
    - Ouvrir l'enregistrement
    - Lire l'enregistrement
    - Vérifier que tout le processus fonctionne sans erreur

2. **Test de robustesse** : Vérifier que l'application gère correctement les cas d'erreur
    - Tester avec des fichiers JSON invalides
    - Tester avec des actions rapides et répétées
    - Vérifier que l'application ne plante pas et affiche des messages d'erreur appropriés
3. **Test de performance** : Vérifier que l'application reste réactive même lors d'une utilisation intensive
    - Jouer de nombreuses notes rapidement
    - Enregistrer une longue séquence

- Vérifier que l'application reste réactive et que les sons sont produits sans délai perceptible