

Rapport individuel - Itération 1 du projet long

- NoteTest.java : Test de la classe Note.java
 - Vérifier que les constructeurs, les setters et les getters fonctionnent avec testConstructorAndGetters(), testSetNoteIndex(), testSetNoteVelocity(), testSetNoteDurationTicks().
 - Vérifier que l'exception *InvalidNoteException* est levée par des notes ne vérifiant pas leurs conditions aux limites.
 - * testValidNoteHighOctave1-2() vérifient que l'on ne peut pas avoir une note d'octave supérieure à 11 (de même avec l'indice de note, l'indice de note MIDI et la vélocité).
 - * testValidNoteLowOctave() vérifie que l'on ne peut pas avoir une note d'octave négative (de même pour les autres indices).
 - * testValidNoteCorrectOctave() vérifie que l'on peut avoir une note d'octave entre 0 et 11 (de même pour les autres indices pour leur encadrement).
- PatternTest.java : Test de la classe Pattern.java
 - Vérifier les constructeur pour créer un pattern vide ou charger un pattern enregistré dans un fichier .mid fonctionnent.
 - * testEmptyConstructorAndGetters(), testMidiConstructorAndGetters().
 - Vérifier qu'ajouter des notes modifie la longueur du pattern.
 - * testAddNoteModifiesPatternLength1-4() vérifient pour un ajout en décalé de 2 mêmes notes d'une même durée, un ajout simultané de 2 mêmes notes d'une durée différente, un ajout simultané de 2 mêmes notes d'une même durée et un ajout en décalé de 2 mêmes notes d'une durée différente.
 - Vérifier que des notes ajoutées (ou retirées) dans un pattern sont retrouvables (ou non) et cohérentes dans son séquenceur.
 - * testAddNoteInTrack1() vérifie que les tracks d'un pattern dans lequel on ajoute une seule note contiennent uniquement le MidiEvent associé à cette dernière.
 - * testAddNoteInTrack2() vérifie que les tracks d'un pattern dans lequel on ajoute plusieurs notes contiennent chacun uniquement le MidiEvent associé à leur note.
 - * testRemoveNoteInTrack1() vérifie que les tracks d'un pattern dans lequel on ajoute plusieurs notes puis enlève une contiennent chacun uniquement le MidiEvent associé à leur note si elle est restante.
 - * testRemoveNoteInTrack2() vérifie qu'une track dont on a enlevé toutes les notes est de longueur nulle.
 - * testRemoveNoteInTrack1() vérifie que les tracks d'un pattern dans lequel on ajoute plusieurs notes puis enlève une quand elle s'est arrêtée contiennent chacun toujours leurs notes respectives.
- MelodyTest.java : Tests vérifiables manuellement.
 - Vérifier la sauvegarde d'un pattern dans un fichier MIDI.
 - * createAndSaveMidi() crée une mélodie d'une dizaine de secondes qui est sauvegardée en tant que test.mid.
 - Vérifier l'ouverture et la lecture d'un pattern sauvegardé dans un fichier MIDI.
 - * Le constructeur Pattern(Path) crée précédemment et la procédure playSequence() permettent de vérifier par l'écoute si le fichier est enregistré et lu correctement.