TAB2MUSIC TESTING DOCUMENT

EECS 2311 SOFTWARE DEVELOPMENT PROJECT

Group 11
March 2022

John Yacoub
Muhammad Sawal
Shaylin Ziaei
Akarshan Kakkar

Table of Contents

1. Testing Methodology

1.1 Objective

The goal of testing our application was to ensure that the main features of the application are working. Testing was conducted for all main methods and covers all major classes implemented. All tests have been done using JUnit Testing. Tests have been conducted using several styles and versions of tab input (for guitar and drums.)

1.2 Derivation

We developed our tests based on testing all features of our application with multiple different inputs to ensure each feature works as expected and that it dealt with edge cases/harder inputs in a correct way.

2. Test Cases

2.1 XMLParser

All the test cases in this class take XML input from a helper method and store it in a string parse. A new Xml parser object is created which parses the string to test the correctness of our parser. There are three test cases for XMLParser, three tests to check the correctness of the number of notes in different measures and three test cases to compare the number of measures of tablature with expected value. These tests have been implemented for both drum and guitar.

2.1.1 testXMLParser1

In this test case, the name of the instrument which is given by the program, is compared with the expected name that is guitar.

2.1.2 testXNumberOfNotes

This test compares the number of notes for the first and second measure which should be 8 and 14 respectively.

2.1.3 testNumberOfMeasures

The number of measures in the tablature is checked to be equal to expected value which is 2 in this test case.

2.1.4-2.1.9

Test cases 2.1.4 - 2.1.9 perform the same tests on a different XML to further check the correctness of our parser.

2.2 XMLParser Sufficiency

The TestParser class checks the correctness of our XMLParser class. It checks that we get the correct instrument name so that the XML is parsed accordingly. It also tests the values of total number of notes and number of notes per measure which are later on needed for parsing.

2.3 Test Guitar Parser

All the test cases in this class take XML input from a helper method and tests are based on the expected values that should be parsed from the XML. XMLParser and GuitarParser objects are created to test the correctness of our guitar parser class. The following XML elements are tested:

2.3.1 Test Strings

This test case compares the values of the Strings that we get from parsing to their actual values in the XML.

2.3.2 Test Frets

This test case compares the values of the frets that we get from parsing to their actual values in the XML.

2.3.3 Test Notes

This test case checks whether or not our parsing returns correct notes in the correct order.

2.3.4 Test NotesLength

This test case checks the length of each note to it’s expected value in the XML.

2.3.5 Test Chords

This test case checks if the correct notes are being played as a chord.

2.3.5 - 2.3.10

Test cases 2.3.5 - 2.3.10 perform the same tests on a different XML to further check the correctness of our parser.

2.4 Test Guitar Parser Sufficiency

The test cases in this class test all the components that are used in playing and visualising the guitar tabs. They check if the information that we get from parsing is correct, and is in the correct order, so that our notes are played and visualised correctly.

2.5 Test Drum Parser

All the test cases in this class take XML input from a helper method and tests are based on the expected values that should be parsed from the XML. XMLParser and DrumParser objects are created to test the correctness of our drum parser class. The following XML elements are tested:

2.5.1 testID

This test checks if the ID of the instrument is equal to the expected ID which is different based on the type of instrument.

2.5.2 testName

The test checks if the name of the instrument is the same as the name that is expected.

2.5.3 testNotes

The values stored in the note list after parsing are compared with their expected values.

2.5.4 testChords

The values that are stored after parsing an array which contains chords are checked with the expected value based on the given tablature.

2.6 Test Drum Parser Sufficiency

The test cases in this class test all the components that are used in playing and visualising the drum tabs. They check if the information that we get from parsing is correct, and is in the correct order, so that our notes are played and visualised correctly.

2.7 Test Play Notes

All the test cases in this class take XML input from a helper method. The class is designed to test the correctness of our JfugueTest and Jfugue for drums classes.

2.7.1 TestPlaying

This test case tests our final output string (contains the notes with alters) which is used for playing. It checks if the value of the string matches to what's expected from the xml.

TestPlaying2 performs the same test using different guitar XML inputs.

TestPlayingDrums performs the same test for drum XML inputs.

2.8 Test Play Notes Sufficiency

The test cases in this class check if the correct notes are marked as altered notes. It also tests if the notes are correct and are in correct order, so that the tabs are played accurately.