## Test Plan Identifier:

Master Test Plan for MuseScore3 SATB\_checker 1.0

## Introduction:

The one test is that it will check if the user has followed the four-part writing rules. There are many rules to four-part harmony. Rules that will be tested will standard rules that composers must adhere to. They include parallel fifth, octave, leading tone resolution, and large leaps. These four rules are basic for the device. \*acc. of checker

## Scope:

The overall goal of this project is to test parallel fifth, parallel octaves, leading tone resolution, and large leaps. Each error that is usually found in four-part writing will receive its own function. For example, parallel fifths will have a checkParallelFifths function to test if the example has a parallel fifth or not. Parallel octaves will have a checkParallelOctave function to test if the example contains a parallel octave or not. Leading tone resolution detection will have a checkLeadingTone function to test if the example contains a leading tone, and if it does, check if that note resolves in the correct direction. Large leap will have a checkLeap function to check if the notes within the same voice have a large interval between them. To utilize these functions, QML will be used to implement the logic in how it will detect errors. HTML and CSS will be used to highlight the errors.

## Test Items:

Since time is limited, this project will be using a plugin to test these functionalities. The source code will be copied and pasted into the MuseScore plugin area and ran from MuseScore. In later development, this may be converted into an app that users may be able to download, exclusively for the harmony checker.

## System Testing:

If the program compiles without compilation errors, then system testing is pass.

## User Acceptance Testing:

Initially, the programmer who constructed the program will be the first one to test his/her code. Once the programmers’ expected output is achieved, he/she will present the program to users who may find interest in the program and will have a use for it in the future. This stage can only be accomplished if the program is operational.

## Features Not to Be Tested:

The display of the notes

This feature is a template that has already been implemented. Logic is correct. Only thing to do is to make it aesthetically pleasing for the user

## Approach:

Load a music file as a test case to show errors, if any. If a specific input is entered in, a specific output should be expected. For example, if parallel fifths are intentionally implemented, the plugin should be able to highlight the error, as well as generate an error message. Testing the product will involve creating one SATB melody. Because of this, user testing is incorporated into it.

## Item Pass/Fail Criteria:

Parallel Fifth test

Parallel Octave test

Input:

Graphical user interface, application, table, Word

Description automatically generated

Expected Output:

Graphical user interface, application, Word

Description automatically generated

Large Leap test

Input:Table, Excel

Description automatically generated

Expected Output:

Table, Excel

Description automatically generated

Leading tone test

TBD

## Suspension Criteria and Resumption Requirements:

List test deliverables, and links to them if available, including the following:

Project Proposal (on GitHub)

Voleres Document (on GitHub)

Test Plan (this document itself)

## Test Environment:

MuseScore3 on Windows

MuseScore4 if fully developed.

## Schedule:

## Tentative Schedule

* 2021
* 2/24 - 4/6 Proposal editing.
* 2/24 – 2/26 Proposal finalization
* 3/3 - 4/6 Requirements Document draft and Final
* (create test plan)
* 5/1- 8/16 Rough Draft of construction of project
* 8/16 - 11/22 Minimal Product Usage
* 2022 (fall)
* 01/03 – 02/03 Implement the test plan.
* 02/04 - 03/05 Analyze Results.
* 03/06 – 11/24 Fix Bugs and Finalize Features
* 11/24 -12/1 Finalize Documentation.
* 12/4 Dress Rehearsal for presentation.
* 12/5 Defense Presentation.

## Responsibilities:

Responsibilities include completing the project in a timely manner, conducting necessary research if roadblocks occur, and meeting with advisor occasionally to provide updates that are a major milestone in the project.

## Risks:

The plugin feature in MuseScore is unstable at times. To prepare for unstable app crashes, frequently save the plugin creator.

## Approvals:

* Dr. Sean Hayes (Project Advisor)
* Steven Yeung (Project Constructor)