

Test Plan

Unit Tests

Test Cases:

Test Case #	Description	Test Data
1.	Check a Piano Key event results in the correct corresponding note (both in pitch and octave)	A4 -> A4
2.	Check that an Octave Up event results in output of one octave higher.	4 -> 5
3.	Check that an Octave Down event results in output of one octave lower.	5 -> 4
4.	Check chord played corresponds to chord recognized. (need to test one of each type of chord, and a few non-chords)	too many
5.	Check that a midi device has been added	midi != null

Proof:

Use case Tests

Test Cases:

Chord Recognition

Actor: User

Requirements: None.

Main Scenario:

1. User plays notes on PolyChord piano
2. System plays the note sounds
3. System looks up notes in database
4. System displays chord to User

Alternatives:

- 1a. User plays only one note
- 1a1. System displays note, instead of Chord.
- 3a. System is unable to load chord database.

3a1. System tells user it could not load the database, please try again later.

4a. Notes do not make a valid chord.

4a1. No chord or note is displayed.

Test Situations:

1. User plays chord
2. User plays single note
3. User plays notes that do not make a chord
4. User plays notes when database is unavailable

Test Coverage:

Situations: 4

Tests: 4

Coverage: 100%

Midi Input

Actor: Advanced User

Requirements: User has MIDI device.

Main Scenario:

1. User plugs in MIDI device

Alternatives:

1a. User does not add MIDI device 1a1. System allows use with keyboard

Test Situations:

1. User plays note on MIDI device

Test Coverage:

Situations: 1

Tests: 1

Coverage: 100%

Metronome

Actor: User

Requirements: User has toggled the metronome on.

Main Scenario:

1. User inputs a bpm value
2. User turns metronome on
3. User hears metronome sound
4. User turns metronome off

Alternatives:

1a. User does not input a value 1a1. System cannot play metronome 1b. User inputs an invalid value 1b1. System behaves incorrectly

Test Situations:

1. User does not enter a value
2. User inputs an invalid value

Test Coverage:

Situations: 2

Tests: 2

Coverage: 100%

Sustain Key

Actor: User

Requirements: None.

Main Scenario:

1. User plays note(s)
2. User presses sustain key
3. System plays corresponding tones.
4. User releases notes (but not sustain key)
5. System continues to play tones
6. User releases sustain key
7. System stops tones.

Alternatives:

1a. User presses sustain key and notes at the same time

1a1. System plays corresponding tones.

1a2. (go to 4.)

6a. User plays more notes.

6a1. System plays additional tones.

6a2. User releases sustain key

6a3. System stops all tones.

Test Situations:

1. Play notes and then sustain & then release notes.
2. Start sustain and notes at the same time.
3. Play notes during sustain.

Test Coverage:

Situations: 3

Tests: 3

Coverage: 100%

Proof:

Acceptance Tests

Test Cases:

1. Test that the application functions as expected on all common* browsers, such as Chrome, Safari, Firefox, and IE. Note that because the MIDI features are browser dependent, a failure message is the expected behavior on other browsers when activating that feature.
2. Test that application recognizes Major, Minor, Dominant, Augmented, Suspended, Half-Diminished, and Diminished chords, as well as their alternative variants.
3. Test that the system accepts MIDI input as expected when using a supported browser.
4. Test that the metronome can play at a default value.
5. Test that the metronome does not accept in invalid bpm number.
6. Test that the system's sustain key acts as expected, sustaining all notes played while it is pressed, and releasing all notes when it is released.
7. Test that the application switches octaves as expected: the octave up button results in the system playing tones an octave higher (within the 5 octave maximum), and the octave down button results in the system playing tones an octave lower (within the 5 octave minimum).

Proof: