**NoteGenerator.test.js:**

“RepeatLastNote should replay the last note”:

* Additional assertions have been added to check if the correct note is being played when repeatLastNote is called.
* The test now uses toHaveBeenNthCalledWith to assert that the console.log function is called with the expected note value.

“Play should handle empty input without throwing errors”:

* Assertions have been added to check if appropriate error messages are logged when empty input is passed to the play function.
* The test now uses toHaveBeenCalledWith to assert that the console.log function is called with the expected error messages.

“Generate should eventually produce each note in the range”:

* The number of iterations has been increased to generator.notes.length \* 1000 to ensure a more thorough coverage of the note range.
* This increases the likelihood that each note in the range will be generated during the test.

**New Test Cases**

“Play should handle invalid note names gracefully”:

* This test ensures that the play function handles invalid note names gracefully and logs an appropriate error message.

“Generate should not produce notes outside the predefined range”:

* This test verifies that the generate function does not produce notes outside the predefined range, even after a large number of iterations.
* It checks that the generated notes do not contain any invalid notes specified in the test.

“Play should handle null or undefined notes gracefully”:

* This test ensures that the play function handles null or undefined notes gracefully and logs appropriate error messages.

“Play should handle special characters or non-alphabetic characters”:

* This test ensures that the play function handles special characters or non-alphabetic characters gracefully and logs appropriate error messages for each invalid note.
* It iterates over an array of invalid notes and checks that the appropriate error message is logged for each note.

“questionManager: questionManager resets userAnswer after calling progressQuestionList()”

* This test ensures that after submitting an answer, the value stored inside of questionManager is reset to zero preventing the user from accidentally submitting the next question before selecting an answer

“'ui: playNoteButton is displayed at start up”

* This test checks that a playNoteButton’s display() function is run at startup, verifying that the note is rendering to the screen

“'ui: playNoteButton is displayed on screen after the previous question is attempted”

* This test checks that a playNoteButton’s display() function is run after a new question is generated, verifying that the note is rendering to the screen.

“'ui: submitButton is displayed at start up”

* This test checks that a submitButton’s display() function is run at startup, verifying that the button for submitting an answer is rendering to the screen

“ui: submitButton submits the answer to questionManager onClick”

* This test checks that after clicking the submitButton, the questionManager will process the answer.

“ui: displayMessageCorrect is called after answering a question correctly”

* This test checks that a message displaying the user answered the question correctly is displayed after an answer has been submitted through the submitButton and checked to be correct with questionManager.

“ui: displayMessageIncorrect is called after answering a question incorrectly”

* This test checks that a message displaying the user answered the question incorrectly is displayed after an answer has been submitted through the submitButton and checked to be incorrect with questionManager.

“ui: a new question is displayed on screen after the previous question is attempted”

* This test checks that a new question is automatically populated and displayed after a previous question is submitted using the submitButton.

“ui: answer 1 is stored in questionManager userAnswer after selection”

* This test checks that answers are stored in questionManager after selecting them from a radio button.

“ui: answer 2 is stored in questionManager userAnswer after initial selection of answer 1 and then selection of answer 2”

* This test checks that answers are overwritten in questionManager after selecting a new answer from a different radio button.