Ashley Chang

504449890

Discussion 1K

Project 3 Report: Guitar Zero

While working on this project, I encountered a number of notable obstacles. One of the main obstacles I faced was moving from one note to another during the translateTune function. Due to the placement of slashes between notes and letters between digits and slashes, I found it difficult to write code that would be able to tell where and when sustained notes started and stopped, as well as how to keep track of what beat number each note occupied. The most difficult part of writing the code was keeping track of and making sure each note had the appropriate number of slashes; this was because I did not realize for a long time that I could manipulate strings and cut out individual segments that the program could check for accuracy. Also, organization proved to be a challenge. Because my code contains so many if-else statements, dealing with brackets and scope was somewhat tricky; sometimes I wrote code in the wrong places and left other parts of the code incomplete.

My project is organized as such: first, the program checks to see whether or not the inputted string is well-formed. It employs an additional function called isColorValid that ensures that any inputted note is a valid note for Guitar Zero (blue, yellow, red, green, orange).

CHECK IF TUNE IS WELL-FORMED

If tune is empty, then well formed

If tune has substance, repeatedly:

If a single slash, mark as beat

If not a slash, check if valid color w/ slash

If not a slash/color, check if digit

Check if number is at most 2 digits long  
 Check if number is followed by valid color

If not a slash/color/digit, return false

Increment to next character in string

Passed all checks, so return true

The next function makes sure the tune is translatable, then writes it out to the instruction string.

CHECK IF TUNE IS TRANSLATABLE

Check if tune is well formed, otherwise return error

Check for beat errors repeatedly:

Check if digit

If digit specifies number less than 2,

Find and record that beat as a badBeat

Check if specified number matches number of slashes

If note length is interrupted, find and record place of interruption as badBeat

Check if tune ends prematurely

If not enough slashes, find and record beat where tune is cut off as badBeat

Move to next character

Tune is translatable. Prepare instructions for translation.

Repeatedly:

Check if digit

Record number of beats to sustain

Add beats to instructions in specified color

Move position to end of sustained note

Check if letter (stand-alone note)

Add color note to instructions

Move position to end of non-sustained note

Check if slash (stand-alone beat)

Add ‘x’ to instructions

Move to next note

Translation complete. Return 0 and leave badBeat unchanged.

List of Test Data

|  |  |
| --- | --- |
| **Test** | **Example(s)** |
| Empty string |  |
| Slashes only | ///// |
| 1 digit | 1  3 |
| 2 digits | 01  05  10 |
| 3 digits | 001  123 |
| Number with slashes | 1/  03/  4////  134/// |
| Color only | R  o  BY  L  Rv |
| Colors with slashes | r/  y//  g/b/  bb/  grybo/  W//B/  B/U/ |
| Number with color and slashes  (single sustained note) including:   * Extra slashes/beats * Correct number of slashes/beats * Single digit notes * Double digit notes * Triple digit notes * Not enough slashes/beats (end prematurely) * Invalid colors | 1r/  01r/  01r//  4b////  5y//  2O//////////  123B//  004R////  14R//////////////  14G///  ///r///  P/3O/// |
| Multiple notes (sustained & nonsustained) with slashes, including:   * Extra slashes/beats between notes * Correct number of slashes/beats * Triple digit notes * Double digit notes * Single digit notes * Single beat notes * End prematurely * Interrupted notes * Invalid colors | R/O/Y/G/B/  3G///b/  g/G/g/B/  r//y/3g///o/  3r///y/o///2B//  001R/B/2G//  01O/2B//  05B/////3R///  1B/2G//  4R////2O/  3r//y/0b//2o/  3r//0b/  ////r////3B///4O////  2Q//4R////o/// |