Honors/AP Computer Science
Design Proposal: Final Project

Project Name: Rote Less, Play More

Developer: Liam Powers

Project Idea

A professional jazz musician today is expected to know hundreds of different standard tunes by memory to be able to play with a group on a gig or jam session when someone calls one of these specific tunes to be played. The task of memorizing these tunes is daunting and arduous, so several aids have been made to exist with this by listing the chords and/or melodies of the tunes in a book or app. However, these aids fail to explicitly mention that many of the tunes have similar or identical chords in certain locations, which is useful for the musician to know, in that learning one tune may mean you're learning large parts of a dozen others. What my project aims to do is to A) allow the user to keep track of what songs they've learned so far, and B) upon entry of the tune, recommend tunes with similar chord progressions to learn next. Ultimately, this should help the musician learn more of these songs and get to higher levels of playing earlier, hence the name of this project: "Rote Less, Play More".

Program Description

_____Rote Less, Play More has two major functions; keeping track of songs the user has/has not learned, and recommending them songs of similar nature upon entry of a certain song title to help them learn the standard tunes quicker. These available options to the user are more obvious under the sample output heading. The program scans a text file containing song titles and their relevant chords in a universal format for the program, allowing it to compare songs across keys with ease. This text file is called "standards.txt" and contains around 50 songs manually entered by me for adequate demonstration of functionality of the recommendation functionality within the timespan. (I like this list. Chord charts here.) For example, when a user asks for this recommendation function, they enter a song, i.e. "Bags' Groove". They then enter a number of consecutive chords in "Bags' Groove" they'd like to check for other songs containing, with a minimum of 3 (1 or 2 could output half the library!) The program looks through the library found in standards.txt and finally outputs song names with that number of the same chords in the same row. The application for the development musician using this program is to then learn the chords of those songs in a resource like *The Real Book* or iReal Pro, which should happen quickly because those songs contain concepts they have already learned in recent memory. Without the application, there are very few resources available that at least list tunes with similar chord structure, nevermind a mathematical way of doing it, so the musician is stuck with stumbling upon similar songs to learn across months and years that they could've easily had under their belt soon after they learned the first one.

The other major functionality of the program is to keep track of tunes the user has learned so far. This is not just an additional courtesy of the program to give the user an easy method of marking what they've learned and what they haven't, but is also handy in helping the recommendation function give better suggestions because it doesn't bother scanning through tunes the user has already marked as learned. The way this feature works is also obvious in the Sample Output header, but there are 4 different functions they can select from the menu; one that lists the song names in standards.txt, one that lets them enter a song they've learned, one that

shows the songs they've learned, and one that shows the songs they haven't learned.

Class & Method Listing

Class	public class ScanTunes	
Fields	public String[][] masterList	2d array of format as follows: (filled in readTunes()) Name Chords Name Chords (continues until tune list is exhausted)
Methods	public void readTunes()	creates a multidimensional array of tunes based on lines in the text file containing names and chords in the usable format, assigns this to the multidimensional array masterList.
Class	public class UserFxns	
Fields	public String[][] learned public String[][] notLearned	learned is initially empty, notLearned is initially equal to masterList; adds to learned and removes from notLearned in learn()
Methods	public String [] repository()	returns the list of songs the program has chords for based on the masterList
	public void learn(String x)	takes specified tune out of notLearned and moves to learned
	public void getLearned()	outputs learned
	public void toLearn()	outputs notLearned
	public void simTunes(int i)	returns songs with similar chords; this will scan through toLearn and give the names of tunes with i (3 or more) chords in a row that match. Looks for the **first occurrence** of a chunk of x chords in the entered song in the **String** of chords in each song, and a positive int returned will add the name of the song to a list to print.
Class	public class Run	
Fields		
Methods	public static void main(String args[])	lets user enter number corresponding to specific function described by methods in the class UserFxns, then does

		those functions; loops until user enters 0

Sample Output

Welcome to the application Rote Less, Play More.

Options:
0) exit
1) output list of tunes held in repository
2) mark a song in repository as learned
3) list songs you haven't learned yet
4) list songs you've learned so far
5) compare a tune you know to find similar tunes you don't

Enter number of desired option:
1
Autumn Leaves
Bags' Groove
(song list continues)

Options:
0) exit
1) output list of tunes held in repository
2) mark a song in repository as learned
3) list songs you haven't learned yet
4) list songs you've learned so far
5) compare a tune you know to find similar tunes you don't

Enter number of desired option:
2
Enter song here: Bags' Groove

Options:
0) exit
1) output list of tunes held in repository
2) mark a song in repository as learned
3) list songs you haven't learned yet
4) list songs you've learned so far
5) compare a tune you know to find similar tunes you don't

Enter number of desired option:
3
Autumn Leaves
(Bags' Groove missing)
(list continues)

Options:

Honors/AP Computer Science
Design Proposal: Final Project

0) exit
1) output list of tunes held in repository
2) mark a song in repository as learned
3) list songs you haven't learned yet
4) list songs you've learned so far
5) compare a tune you know to find similar tunes you don't

Enter number of desired option:
4
Bags' Groove

Options:
0) exit
1) output list of tunes held in repository
2) mark a song in repository as learned
3) list songs you haven't learned yet
4) list songs you've learned so far
5) compare a tune you know to find similar tunes you don't

Enter number of desired option:
5
Enter name of tune: Bags' Groove
Enter number of chords to compare (minimum of 3): 3
Similar tunes: (lists tunes, Autumn Leaves missing)