Huffman’s Song

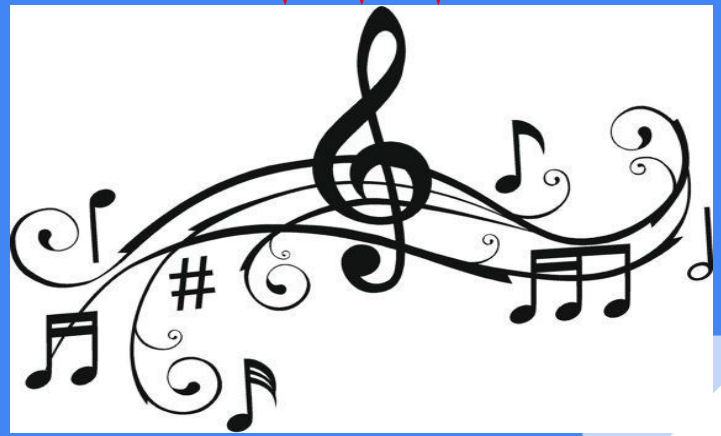
CS 345 Assignment #5

```
var x = Compress code "var x = 0; x += 23; print x"

Initialize the interpreters
Initialize tracks 0 to 4

Interpret code x on 0 through 0
Play track 0

Wait until input
```



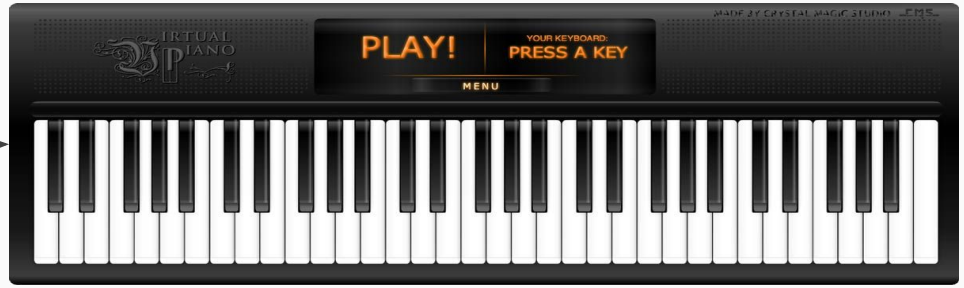
What is it?

- DSL that converts any code/string into a Huffman tree
- In the process creates another encoded file for making music
- Uses “interpreters” to transform encoded file into one of many different styles of songs
- Song and Huffman coding outputted to a file

Music Theory:

- Takes advantage of the fact that character occurrence in a file is like note occurrence in song--some more frequent than others
- Also, only a discrete set to choose from (keys on computer and musical keyboard)

Think of it like a robotic keyboard player that uses sets of keys to write songs!



Music Theory: Interpreters

- Once Huffman process has counted the number of occurrences of each character in a file, they are ranked by frequency
- We then map each character to its frequency rank in the file (and break ties)
- Ex: example.txt = “aaabbc” → [(a, 0), (b, 1), (c, 2)]
- a is most frequent letter so it gets the first spot! “tonic” pitch

Music Theory: Interpreters

- Each interpreter has a mapping from the rank of the frequency of character occurrence to the pitches more likely to occur in a particular style of song
- Duration is considered as well, with more common characters lasting for longer durations

Challenges

- Using functional programming
- Setting up our IDE’s
- Grasping the syntax of Scala and the javax library
- Merging our work together

Demo