Triads e Pepe

What this talk is About

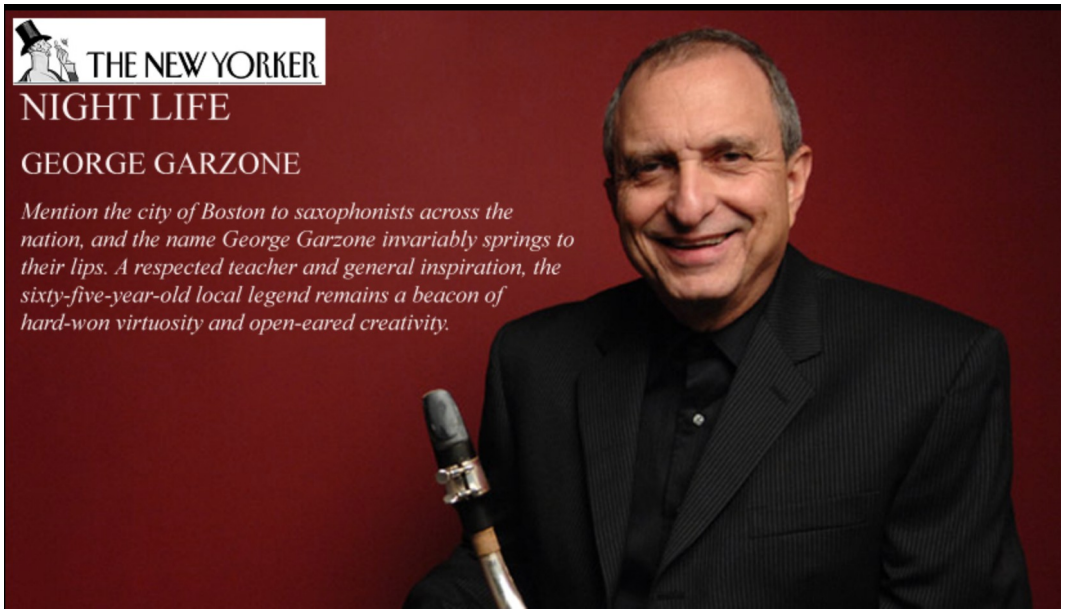
- Use KDB+/Q for generating jazz music (lines)
- based on George Garzone's concepts
- Inputs are a few musical parameters
  - Starting note
  - Triad Quality
  - Step
  - How many triads in the output
- outputs midi files

Who I Am

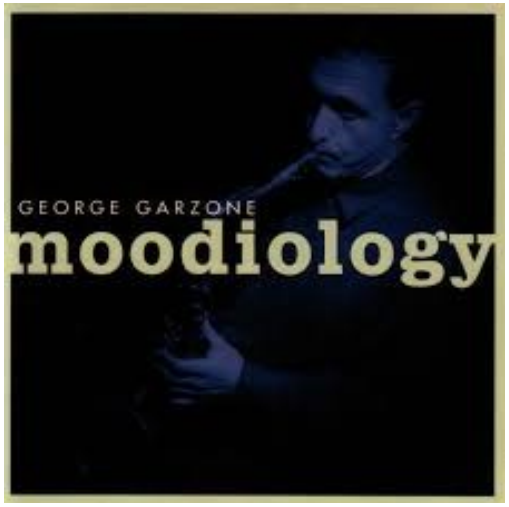
- Steve Wirts
  - Software developer 25 years in NYC area
  - KDB/Q+, Java, Javascript, Smalltalk
  - Currently at Virtu

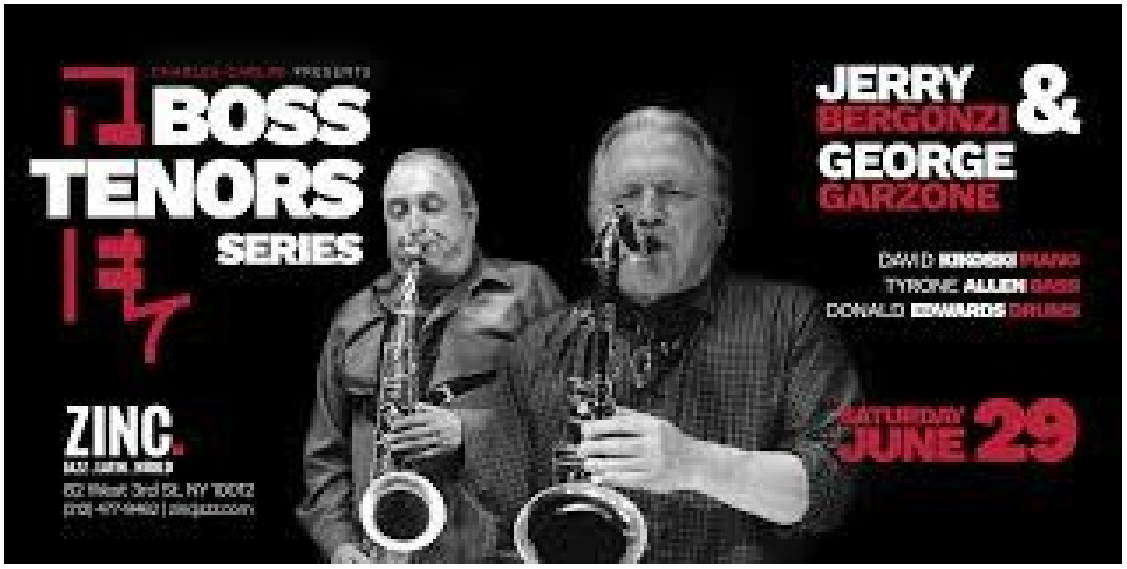
George Garzone

- elite world renown jazz musician (saxophone)
- currently faculty at Berkelee
- historically significant jazz innovator



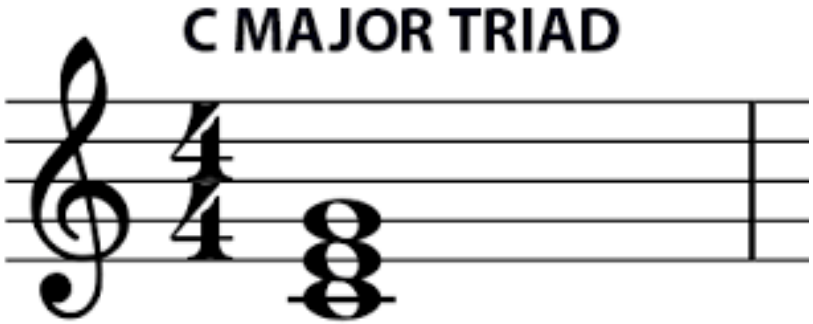
George Garzone





Triads e Pepe?

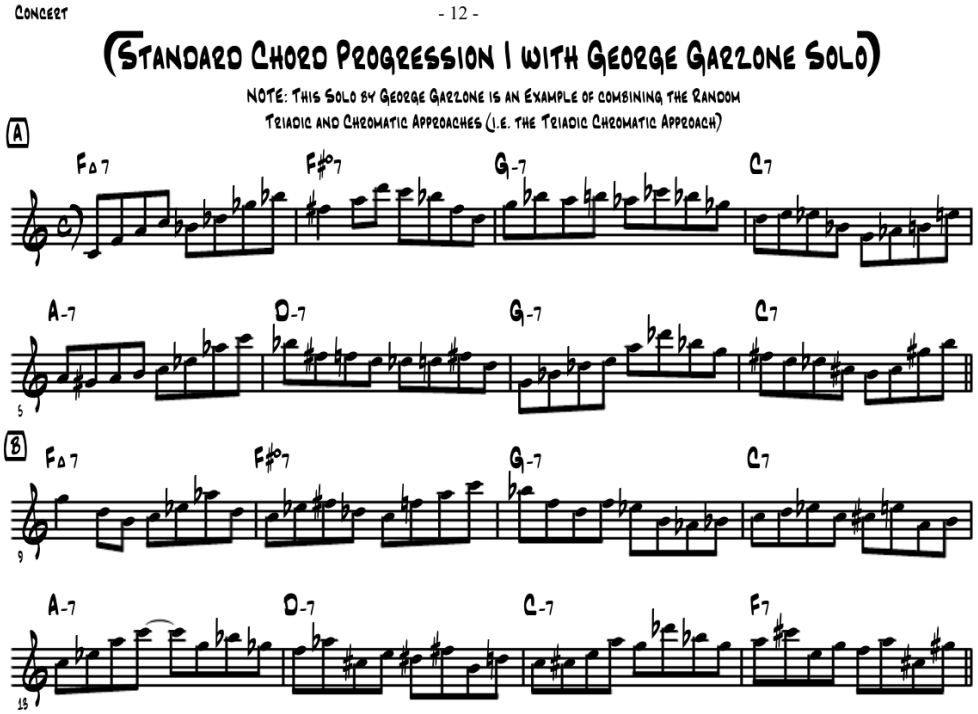
- only 3 ingredients

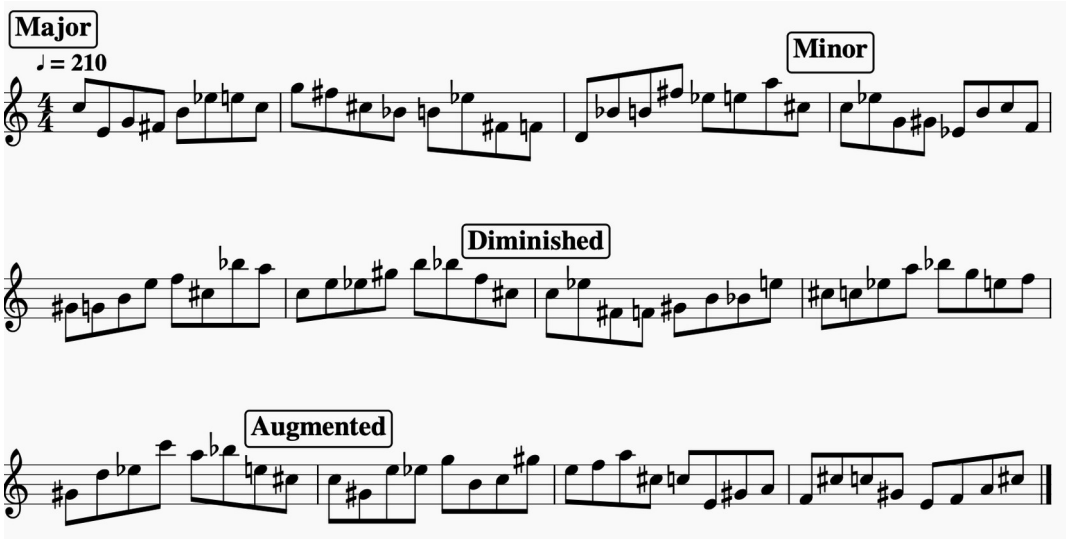


Cacio e Pepe

- deceptively difficult to get just right

The “Sound”





*play both examples now

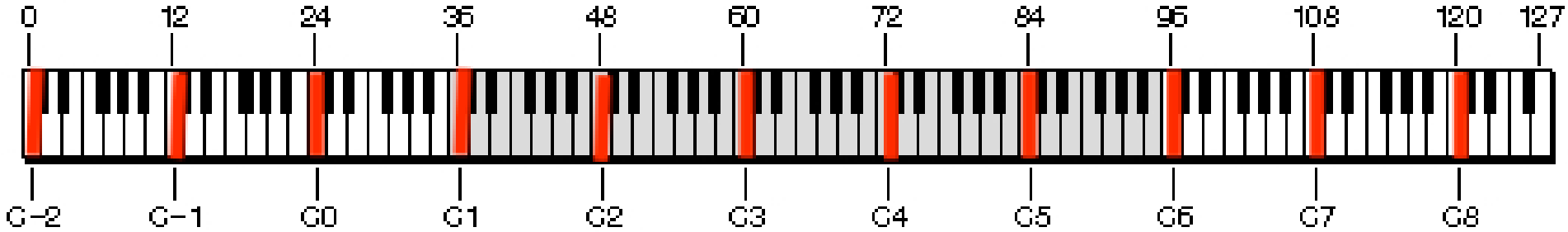
Towards an Algorithm

- George developed this approach
  - Over time starting in 1990s
  - Listening to Coltrane
  - Spark happened at an ear training class at NEC

- Triadic Chromatic Approach
  - Connect random triads of same quality
  - Starting note offset by ½ step from previous note
  - Avoid patterns
    - *don’t repeat inversions*

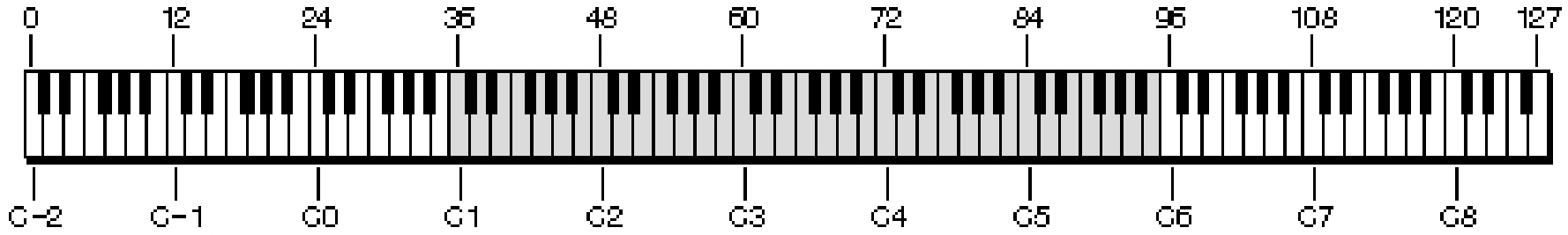
Music Theory for Computer People

- -octave is a scientific term
- same note just “higher” sounding
  - each octave is 2 * frequency of the next below
    - C3 130.81Hz
    - C2 65.41Hz
    - C1 32.70Hz
    - C0 16.35Hz



Music Theory for Computer People

- Musical Instrument Digital Interface (midi)
  - semitone (half step) is the atomic unit
  - an octave has 12 semitones, music is base 12
  - notes are integers between the audible lowest and highest notes

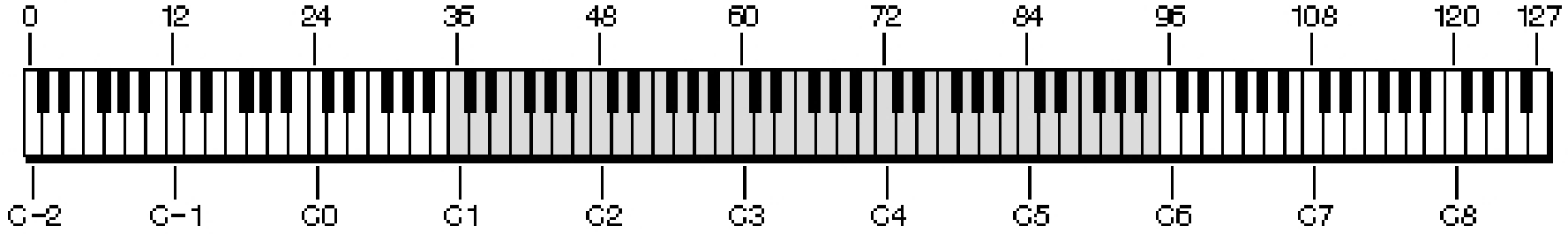


Music Theory for Computer People

- Musical Instrument Digital Interface (midi)
  - C is the name of the lowest note

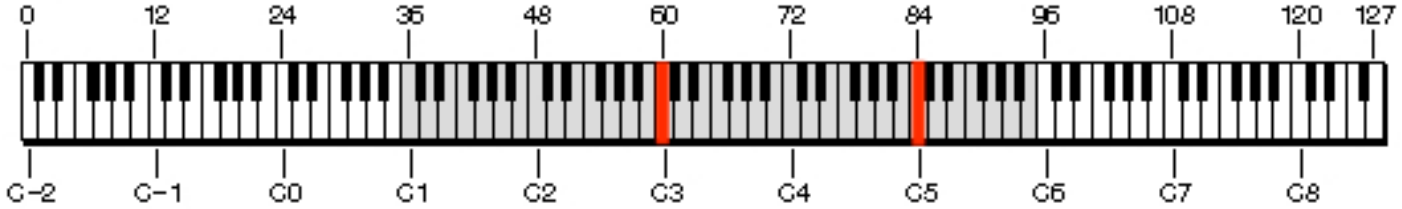
1 octave ((#)sharp (b)flat – ebony keys)

- 0 1 2 3 4 5 6 7 8 9 10 11 12
- C (C# Db) D (D# Eb) E F (F# Gb) G (G# Ab) A (A# Bb) B C



Music Theory for Computer People

- A Quick note about range
  - bosendofer imperial piano has 8.5 octaves 0 - 102
  - midi accommodates 0 - 128
  - western music, a practical range is 60 – 84 (0-24)
    - 2 octaves easy to read in sheet music, easy to play on saxophone

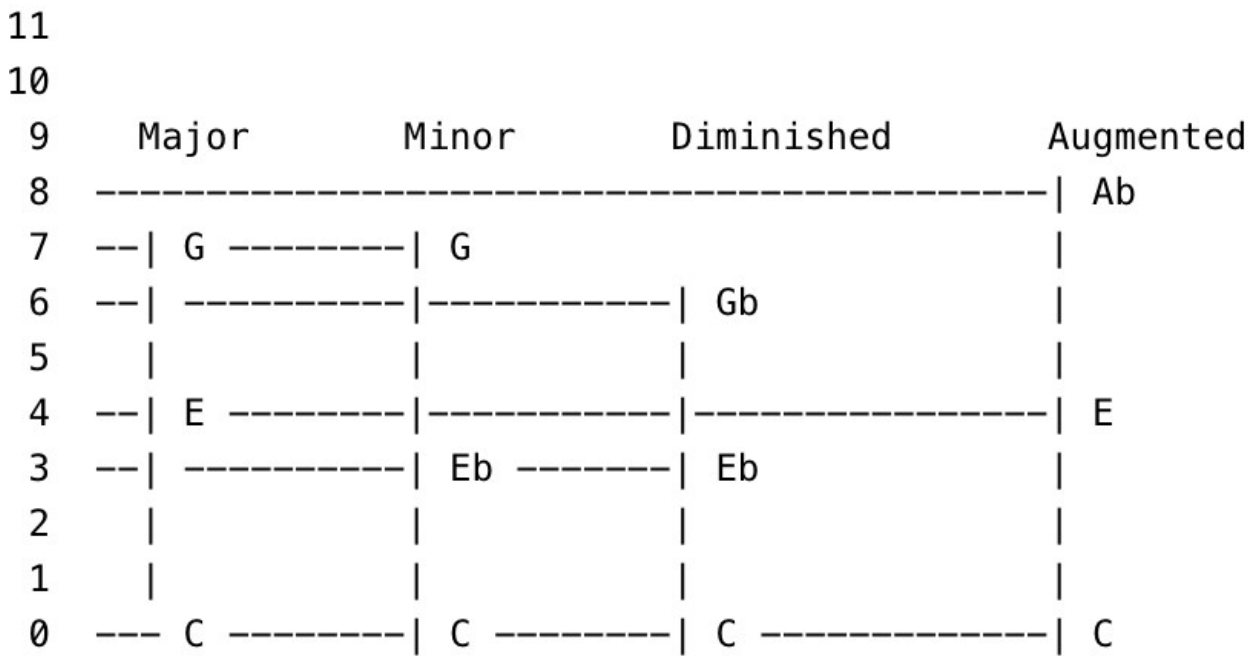


Music Theory for Computer People

- an interval is a number of semitones (a distance) between 2 notes
- a triad is a cluster of 3 notes - root, middle, top (2 stacked intervals)
- 4 basic triad “Qualities”

Music Theory Triads

quality	music notes	midi note values	interval sizes
C Major	C E G	0 4 7	4 3
C Minor	C Eb G	0 3 7	3 4
C Diminished	C Eb Gb	0 3 6	3 3
C Augmented	C E Ab	0 4 8	4 4



Music Theory – the Inversion

- an inversion is when the notes of a triad are rearranged with respect to which one is on bottom
- a triad has 3 notes so it has 3 possible inversions
- three stacked inversions with the root as the middle note is called an INVERSION LADDER
  - there are 5 notes in an inversion ladder
- Critical to the epiphany

```
4    top                  G
3    middle          E    E
2    root       C    C    C – 1st inversion root middle top
1    top        G    G – 3nd inversion (top down an octave) root middle
0    middle     E – 2nd inversion (middle down an octave) (top down an octave) root
```

so

```
3rd inversion   0 1 2
2nd inversion   1 2 3
1st inversion   2 3 4
```

Computer Theory for Music People

- computer people count things starting from 0

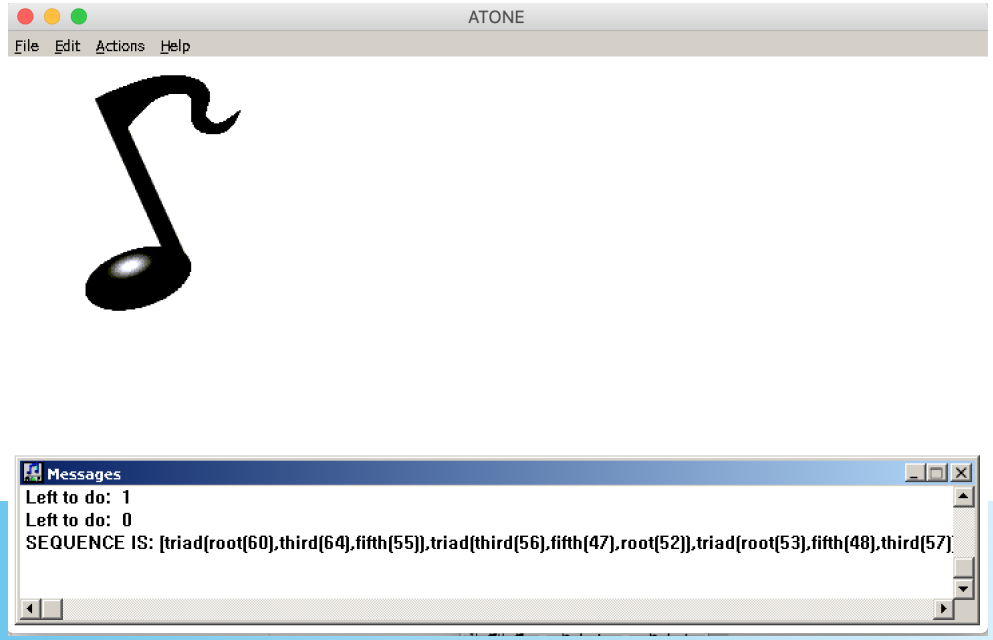
Try and Fail over the Years

England prolog version – late 1990's

Java version early 2000's

Finally got the algorithm right do it in KDB+/Q

- Q God Steve Apter: "get to the tiniest version of the algorithm that you can – you have to think really hard for awhile "



The Epiphany

- YOU CAN REPEAT INVERSIONS!
- there are 2 keys to avoiding hearing repetition, inversion and order of the notes played within
  - anything is trivial once it becomes obvious

The Algorithm

Imaged by Heritage Auctions, HA.com

The Algorithm

- construct music "lines" by connecting random triads of the same quality
- random means to select at random an inversion and a note sequence
- * you can repeat an inversion but note ordering within the inversion must be different from the previous triad
- consecutive triads start on a note that is up or down 1 semitone from last note of the previous triad
- if the last note of a triad is G, the next triad must start on a Gb(F#) or a Ab(G#)
- these notes (Gb or Ab) can be the root middle or top of the next triad

The Algorithm

- there are 18 possible note sequences for 1 triad of any quality, these are called
- INVERSION LADDER SEQUENCE INDEXES

2nd inversion	3rd inversion	1st inversion
0 1 2	1 2 3	2 3 4
0 2 1	1 3 2	2 4 3
1 0 2	2 1 3	3 2 4
1 2 0	2 3 1	3 4 2
2 0 1	3 1 2	4 2 3
2 1 0	3 2 1	4 3 2

The Algorithm - example

- TCA algorithm inputs (reasonable defaults in parens)
  - 1) starting note (72) middle c
  - 2) a triad (4 3)
    - major "happy" sounding triad expressed as 2 stacked interval sizes
  - 3) count of repeating triads per output line (8)
  - 4) the (step) size (1 semitone) between successive triads
    - up or down at random
  - 5) high low bounds (60 84)
    - any line that goes above or below this range is disqualified

The Algorithm – example (1)

```
1) select a triad quality, lets say major 4 3
    compute the inversion ladder
        ladder:{(0;x 1;12-x 0;12;12+x 1)}
        tq:4 3
        il:ladder tq
            4 top       12 + tq[1]          = 15
            3 middle    12                  = 12
            2 root      12 - tq[0]          =  8
            1 top       tq[1]               =  3
            0 middle    0                   =  0
```

The Algorithm – example (2)

```
2) compute the inversion ladder sequence indexes
    ilsi:raze(0 1 2;1 2 3;2 3 4)@\:(0 1 2;0 2 1;1 0 2;1 2 0;2 1 0;2 0 1)

      0 1 2
      0 2 1
      1 0 2
      1 2 0
      2 1 0
      2 0 1
      1 2 3
      1 3 2
      2 1 3
      2 3 1
      3 2 1
      3 1 2
      2 3 4
      2 4 3
      3 2 4
      3 4 2
      4 3 2
      4 2 3
```

*also: raze til[3]+\:(0 1 2;0 2 1;1 0 2;1 2 0;2 1 0;2 0 1)

The Algorithm – example (3)

```
3) apply the 18 inversion ladder sequence indexes to our inversion ladder il (0 3 8 12 15)
    il @ ilsi

        0  3  8
        0  8  3
        3  0  8
        3  8  0
        8  0  3
        8  3  0

        3  8  12
        3  12 8
        8  3  12
        8  12 3
        12 3  8
        12 8  3

        8  12 15
        8  15 12
        12 8  15
        12 15 8
        15 8  12
        15 12 8
```

The Algorithm – example (4)

```
4) compute the INVERSION LADDER INTERVALS from this
    ili:{1_ deltas x} each il@ilsi

       3  5
       8  -5
       -3 8
       5  -8
       -5 -3
       -8 3

       5  4
       9  -4
       -5 9
       4  -9
       -4 -5
       -9 5

       4  3
       7  -3
       -4 7
       3  -7
       -3 -4
       -7 4
```

The Algorithm – example (5)

5) a random triad is now (-1 or 1) and one of the inversion ladder interval from (4)

```
1 7 -3
   or
-1 7 -3
```

The Algorithm – example (6)

```
6) repeat (4), 8 times without repeating the same inversion ladder interval
        flip (enlist 8?1 -1),flip -8 ? ili

           -1 -5 9
           1  -3 -4
           1  4  3
           1  -9 5
           -1 -7 4
           1  5  -8
           -1 3  -7
           1  9  -4
```

The Algorithm – example (7)

```
7) raze and scan the sums, replace the first value with starting note value
     sums 72, 1 _ raze flip (enlist 8?1 -1),flip -8 ? ili

       72 67 76 77 74 70 71 75 78 79 70 75 74 67 71 72 77 69 68 71 64 65 74 70
```

The Algorithm - solution

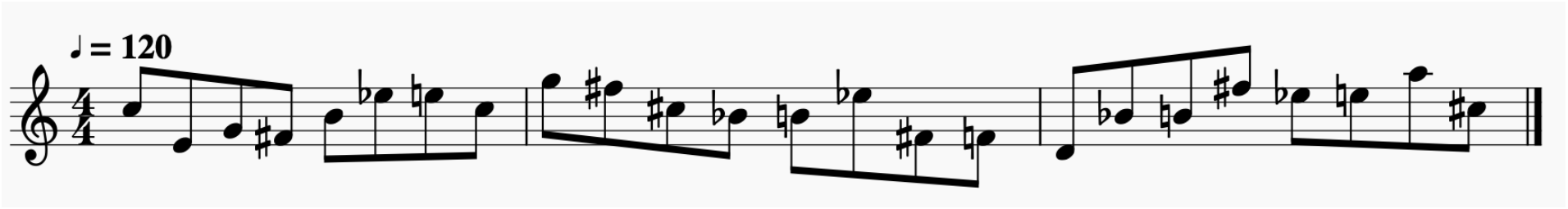
```
triads:(!). flip (
 (`major;4 3);
 (`minor;3 4);
 (`diminished;3 3);
 (`augmented;4 4)
 );

generateline:{sums 72,1 _ raze flip (enlist 8?1 -1),flip -8 ? {1_ deltas x} each ladder[triads x]@ilsi}
```

The Algorithm – wrap up

```
8) save to disk as a midi file
    `:line.midi 1: midi generateline`major

9) reject for range and ending note, repeat a zillion times
```



The Algorithm – wrap up

- Q is a general purpose language
- Try doing small/odd things with it in your free time
  - Its fun!!
- You can do more things with it than you think

- Links
  - https://github.com/stevewirts/kdb-midi-generation
  - https://github.com/stevewirts/kdb-tca-talk

Fun Time!!!

- Lets try some generated lines with George now!