Task 3 - Mutation

About the task …

This task establishes the mutation functionality of this project. Given a music sample, mutation is defined as the following: changing the pitch and/or duration of a random note in each of the three melodies. In the future, it may be interesting to only mutate some of the melodies as opposed to all. This design, however, optimizes the most variation for more significant user interaction (i.e. minimize the number of generations needed to produce a sample).

Demo

```
Melody1 mutation
Melody2 mutation
Melody3 mutation

[2]> ( demo--mutation )
Melody 1: G' G/2' F2' C' A' C/2' G2' B' C/2' C2' F2' F2' B' C/2'
E/2' G2' A/2' C' D' B' D2' G'
Melody 2: B' C' F2' G' C' F2' D2' G2' F2' B' B' C/2' C/2' C/2'
G/2' G' C2' E/2' G2' A/2' D' A'
Melody 3: D2, F, A4, F, G, D4, A, E4, C, B4, F2, D,

-Mutation-
Melody 1: G' G/2' F2' C' A' C/2' G2' B' C/2' D' F2' F2' B' C/2'
E/2' G2' A/2' C' D' B' A/2' G'
Melody 2: B' C' F2' G' C' F2' A/2' G2' F2' B' B' C/2' C/2' C/2'
G/2' G' D' E/2' G2' A/2' D' A'
Melody 3: D2, F, A4, F, G, D4, A, E4, C, F4, F2, D,

------------------Trial 2------------------
Melody 1: G' G/2' F2' C' A' C/2' G2' B' C/2' D' F2' F2' B' C/2'
E/2' G2' A/2' C' D' B' A/2' G'
Melody 2: B' C' F2' G' C' F2' A/2' G2' F2' B' B' C/2' C/2' C/2'
G/2' G' D' E/2' G2' A/2' D' A'
Melody 3: D2, F, A4, F, G, D4, A, E4, C, F4, F2, D,

-Mutation-
Melody 1: A2' G/2' F2' C' A' C/2' G2' B' A2' D' F2' F2' B' C/2'
E/2' G2' A/2' C' D' B' A/2' G'
```

```
Melody 2: B' C' F2' A2' C' F2' A/2' G2' F2' B' B' A2' C/2' C/2'
G/2' G' D' E/2' G2' A/2' D' A'
Melody 3: D2, F, A4, F, G, D4, A, E4, C, C2, F2, D,

------------------Trial 3------------------
Melody 1: A2' G/2' F2' C' A' C/2' G2' B' A2' D' F2' F2' B' C/2'
E/2' G2' A/2' C' D' B' A/2' G'
Melody 2: B' C' F2' A2' C' F2' A/2' G2' F2' B' B' A2' C/2' C/2'
G/2' G' D' E/2' G2' A/2' D' A'
Melody 3: D2, F, A4, F, G, D4, A, E4, C, C2, F2, D,

-Mutation-
Melody 1: A2' G/2' G' C' A' C/2' G2' B' A2' D' F2' F2' B' C/2'
E/2' G2' A/2' C' D' B' A/2' G'
Melody 2: B' C' G' A2' C' F2' A/2' G2' F2' B' B' A2' C/2' C/2'
G/2' G' D' E/2' G2' A/2' D' A'
Melody 3: D2, F, A4, F, G, F2, A, E4, C, C2, F2, D,

------------------Trial 4------------------
Melody 1: A2' G/2' G' C' A' C/2' G2' B' A2' D' F2' F2' B' C/2'
E/2' G2' A/2' C' D' B' A/2' G'
Melody 2: B' C' G' A2' C' F2' A/2' G2' F2' B' B' A2' C/2' C/2'
G/2' G' D' E/2' G2' A/2' D' A'
Melody 3: D2, F, A4, F, G, F2, A, E4, C, C2, F2, D,

-Mutation-
Melody 1: A2' G/2' G' C' A' C/2' G2' B' G2' D' F2' F2' D2' C/2'
E/2' G2' A/2' C' D' B' A/2' G'
Melody 2: B' C' G' A2' C' F2' A/2' G2' F2' D2' B' G2' C/2' C/2'
G/2' G' D' E/2' G2' A/2' D' A'
Melody 3: D2, F, A4, F, G, F2, A, E4, C, C2, F4, D,

NIL
[3]> ( demo--mutation )
Melody 1: G2 A2 E2 B2 A2 A2 C/2 F/2 D/2 G/2 A2 B2 G/2 B/2 F/2 G
E D/2 D B B/2 A/2 D/2 C/2
Melody 2: D' A2' A' F' B' B2' E2' A2' C' G' C2' G2' D' G/2' B2'
D/2' F2' F2'
Melody 3: C2, C2, B2, A2, F, A4, G2, G4, G, D2, F2, F2,

-Mutation-
```

Melody 1: G2 A2 E2 B2 A2 A2 C/2 F/2 D/2 G/2 A2 B2 G/2 C/2 F/2 G E D/2 D B B/2 A/2 D/2 C/2
Melody 2: D' A/2' A' F' B' B2' E2' A2' C' G' C2' G2' D' G/2' B2' D/2' F2' F2'
Melody 3: C2, C2, C4, A2, F, A4, G2, G4, G, D2, F2, F2,

------------------Trial 2------------------
Melody 1: G2 A2 E2 B2 A2 A2 C/2 F/2 D/2 G/2 A2 B2 G/2 C/2 F/2 G E D/2 D B B/2 A/2 D/2 C/2
Melody 2: D' A/2' A' F' B' B2' E2' A2' C' G' C2' G2' D' G/2' B2' D/2' F2' F2'
Melody 3: C2, C2, C4, A2, F, A4, G2, G4, G, D2, F2, F2,

-Mutation-
Melody 1: G2 A2 E2 B2 A2 C2 C/2 F/2 D/2 G/2 A2 B2 G/2 C/2 F/2 G E D/2 D B B/2 A/2 D/2 C/2
Melody 2: D' A/2' A' F' B' B2' E2' A2' C' G' C2' G2' D' G/2' B2' D/2' F' F2'
Melody 3: C2, C2, C4, A2, F, A4, G2, G4, G, D2, F2, C4,

------------------Trial 3------------------
Melody 1: G2 A2 E2 B2 A2 C2 C/2 F/2 D/2 G/2 A2 B2 G/2 C/2 F/2 G E D/2 D B B/2 A/2 D/2 C/2
Melody 2: D' A/2' A' F' B' B2' E2' A2' C' G' C2' G2' D' G/2' B2' D/2' F' F2'
Melody 3: C2, C2, C4, A2, F, A4, G2, G4, G, D2, F2, C4,

-Mutation-
Melody 1: G2 A2 E2 B2 A2 C2 C/2 F/2 D/2 G/2 A2 B2 G/2 C/2 D G E D/2 D B B/2 A/2 D/2 C/2
Melody 2: D' A/2' A' F' B' B2' E2' A2' C' C2' C2' G2' D' G/2' B2' D/2' F' F2'
Melody 3: F4, C2, C4, A2, F, A4, G2, G4, G, D2, F2, C4,

------------------Trial 4------------------
Melody 1: G2 A2 E2 B2 A2 C2 C/2 F/2 D/2 G/2 A2 B2 G/2 C/2 D G E D/2 D B B/2 A/2 D/2 C/2
Melody 2: D' A/2' A' F' B' B2' E2' A2' C' C2' C2' G2' D' G/2' B2' D/2' F' F2'
Melody 3: F4, C2, C4, A2, F, A4, G2, G4, G, D2, F2, C4,

```
-Mutation-
Melody 1: G2 A2 E2 B2 A2 C2 C/2 F/2 E G/2 A2 B2 G/2 C/2 D G E
D/2 D B B/2 A/2 D/2 C/2
Melody 2: D' A/2' A' F' B' B2' E2' A2' D/2' C2' C2' G2' D' G/2'
B2' D/2' F' F2'
Melody 3: F4, C2, C4, A2, E4, A4, G2, G4, G, D2, F2, C4,

NIL
[4]>
```

Demo Code

```
; Method for demoing mutation on a music sample
;   -Mutation is defined as changing the pitch and duration of a note in
each of the melodies of
;   a music sample
;   -This method generates a music sample and performs mutation 4 times.
( defmethod demo--mutation ()
    ( setf melody1-notes ( generate-melody1 ) )
    ( setf melody2-notes ( generate-melody2 melody1-notes ) )
    ( setf melody3-notes ( generate-bassline ) )

    ( setf sample
        ( make-instance 'music
            :melody1 melody1-notes
            :melody2 melody2-notes
            :melody3 melody3-notes
            :str-representation ""
        )
    )
    ( display-all-melodies sample )
    ( mutation sample )
    ( terpri )
    ( format t "-Mutation-~%" )
    ( display-all-melodies sample )
    ( terpri )
    ( format t "-------------------Trial 2--------------------~%")
    ( display-all-melodies sample )
    ( mutation sample )
    ( terpri )
```

```
    ( format t "-Mutation-~%" )
    ( display-all-melodies sample )
    ( terpri )
    ( format t "--------------------Trial 3--------------------~%")
    ( display-all-melodies sample )
    ( mutation sample )
    ( terpri )
    ( format t "-Mutation-~%" )
    ( display-all-melodies sample )
    ( terpri )
    ( format t "--------------------Trial 4--------------------~%")
    ( display-all-melodies sample )
    ( mutation sample )
    ( terpri )
    ( format t "-Mutation-~%" )
    ( display-all-melodies sample )
    ( terpri )

)
```

Code

```
; Method for mutating a music sample -- each melody is mutated

( defmethod mutation ( ( m music ) )
    ( setf melody1-list ( music-melody1 m ) )
    ( setf melody2-list ( music-melody2 m ) )
    ( setf melody3-list ( music-melody3 m ) )

    ( mutate-note-list melody1-list *CMAJOR* *melody-durations* )
    ( mutate-note-list melody2-list *CMAJOR* *melody-durations* )
    ( mutate-note-list melody3-list *CMAJOR* *bassline-durations* )

)

; Helper method for mutating an individual notes list -- the mutation
; changes the pitch and duration of a single note
( defmethod mutate-note-list ( ( notes-list list ) ( pitch-list list ) (
duration-list list ) )
```

```
    ( setf note-to-change ( nth ( random ( length notes-list ) ) )
notes-list ) )
    ( setf new-note-pitch ( nth ( random ( length pitch-list ) ) )
pitch-list ) )
    ( setf new-note-duration ( nth ( random ( length duration-list ) ) )
duration-list ) )
    ( setf new-note-str ( generate-str-representation new-note-pitch
new-note-duration ( note-octave note-to-change ) ) )

    ; if the same note is generated, recursively call on the method again
    (cond
        ( ( and ( equal new-note-pitch ( note-pitch note-to-change ) )
                ( equal new-note-duration ( note-pitch note-to-change ) )
)
            ( mutate-note-list notes-list pitch-list duration-list )
        )
    )

    ( setf ( note-pitch note-to-change ) new-note-pitch )
    ( setf ( note-duration note-to-change ) new-note-duration )
    ( setf ( note-str-representation note-to-change ) new-note-str )

)
```