Task 13 - Finish GA Implementation

This is the task where the genetic algorithm is “finished.” Multiple generations of music samples can be run through. I conclude the task with my thoughts in the final section.

Demo

```
[2]> (ga)


Generation 0 population ...

1     #<MUSIC #x1ABAB47D>  0
2     #<MUSIC #x1ABAB98D>  0
3     #<MUSIC #x1ABABC09>  0
4     #<MUSIC #x1ABABE35>  0
5     #<MUSIC #x1ABACC2D>  0
6     #<MUSIC #x1ABAD009>  0
7     #<MUSIC #x1ABAD195>  0
8     #<MUSIC #x1ABAD3A5>  0
9     #<MUSIC #x1ABCF0F1>  0
10    #<MUSIC #x1ABEBAC1>  0

----------------Selection----------------
X:1
T:Selection
C:Dystopian Tuesday
M:4/4
L:1/4
Q:1/4=120
V:S clef=treble name=Melody1 snm=Melody1
V:A clef=treble name=Melody2 snm=Melody2
%%score [ ( S ) ( A ) ]
K:C
%%MIDI program 0
V:S
D' E' G' D' B2' B' B' z4 |]
V:A
D E G D B2 B B z4 |]
```

```
V:S
A A G/2 B2 G/2 G F B z4 |]
V:A
A2 E2 C/2 D G G C/2 z4 |]
V:S
A/2' D/2' D2' F2' A/2' E2' F/2' z4 |]
V:A
A, A2, B, A, B2, B, z4 |]
[Sample 0] Melody 1 ranking (out of 10)? 3
Melody 2 ranking (out of 10)? 4

[Sample 1] Melody 1 ranking (out of 10)? 2
Melody 2 ranking (out of 10)? 1

[Sample 2] Melody 1 ranking (out of 10)? 0
Melody 2 ranking (out of 10)? 0

average fitness of population 1 = 0.65


Generation 1 population ...

1     #<MUSIC #x1ABB12ED>  0
2     #<MUSIC #x1ABB1329>  0
3     #<MUSIC #x1ABB1365>  3
4     #<MUSIC #x1ABB1395>  0
5     #<MUSIC #x1ABB27E5>  0
6     #<MUSIC #x1ABB2841>  7/2
7     #<MUSIC #x1ABB287D>  0
8     #<MUSIC #x1ABB28B9>  0
9     #<MUSIC #x1ABB2901>  0
10    #<MUSIC #x1ABC1685>  0

----------------Selection----------------
X:1
T:Selection
C:Dystopian Tuesday
M:4/4
L:1/4
Q:1/4=120
V:S clef=treble name=Melody1 snm=Melody1
```

```
V:A clef=treble name=Melody2 snm=Melody2
%%score [ ( S ) ( A ) ]
K:C
%%MIDI program 0
V:S
F2 A G/2 B2 G/2 G F B z4 |]
V:A
A2 E2 C/2 E G G C/2 z4 |]
V:S
D F D/2 F2 E' A/2' E2' F/2' z4 |]
V:A
F B/2 F/2 A2 B/2 G B2 F/2 z4 |]
V:S
F2 A G/2 B2 G/2 G F B z4 |]
V:A
A2 E2 C/2 E G G C/2 z4 |]
[Sample 0] Melody 1 ranking (out of 10)? 24

[ERROR] A ranking must be a natural number x such that -1 < x <
11.
Melody 1 ranking (out of 10)? 3
Melody 2 ranking (out of 10)? 2

[Sample 1] Melody 1 ranking (out of 10)? 4
Melody 2 ranking (out of 10)? 1

[Sample 2] Melody 1 ranking (out of 10)? 0
Melody 2 ranking (out of 10)? 2

average fitness of population 2 = 3.15


Generation 2 population ...

1     #<MUSIC #x1ABB232D>  5
2     #<MUSIC #x1ABB2369>  2
3     #<MUSIC #x1ABB238D>  5
4     #<MUSIC #x1ABB23BD>  5
5     #<MUSIC #x1ABB26D5>  5/2
6     #<MUSIC #x1ABB2731>  5/2
7     #<MUSIC #x1ABB276D>  1
```

```
8     #<MUSIC #x1ABB27C9>  5/2
9     #<MUSIC #x1ABB2811>  5
10    #<MUSIC #x1ABBBA1D>  1

----------------Selection----------------
X:1
T:Selection
C:Dystopian Tuesday
M:4/4
L:1/4
Q:1/4=120
V:S clef=treble name=Melody1 snm=Melody1
V:A clef=treble name=Melody2 snm=Melody2
%%score [ ( S ) ( A ) ]
K:C
%%MIDI program 0
V:S
A/2' C' F' B2' F' C D2 z4 |]
V:A
A2 E2 C/2 E A2 F C/2 z4 |]
V:S
F G/2 A' F/2' D2' F2' F' z4 |]
V:A
D/2 D2' F' F2' C/2' z4 |]
V:S
F G/2 D/2 F2 D/2' C2' D' z4 |]
V:A
D/2 F/2 F/2 E2 F/2 G B2 z4 |]
[Sample 0] Melody 1 ranking (out of 10)? 3
Melody 2 ranking (out of 10)? 4

[Sample 1] Melody 1 ranking (out of 10)? 7
Melody 2 ranking (out of 10)? 1

[Sample 2] Melody 1 ranking (out of 10)? 2
Melody 2 ranking (out of 10)? 9

average fitness of population 3 = 6.4


Generation 3 population ...
```

```
1     #<MUSIC #x1ABB1EB5>  7
2     #<MUSIC #x1ABB1EF1>  5
3     #<MUSIC #x1ABB1F2D>  11
4     #<MUSIC #x1ABB1F5D>  5
5     #<MUSIC #x1ABB2255>  4
6     #<MUSIC #x1ABB22B1>  17/2
7     #<MUSIC #x1ABB22ED>  5
8     #<MUSIC #x1ABB2329>  5
9     #<MUSIC #x1ABB2391>  9/2
10    #<MUSIC #x1ABDFC75>  9

----------------Selection----------------
X:1
T:Selection
C:Dystopian Tuesday
M:4/4
L:1/4
Q:1/4=120
V:S clef=treble name=Melody1 snm=Melody1
V:A clef=treble name=Melody2 snm=Melody2
%%score [ ( S ) ( A ) ]
K:C
%%MIDI program 0
V:S
F/2 F/2 G C2 D/2' E' D' z4 |]
V:A
D F/2 D/2 F2 C G B/2 z4 |]
V:S
A/2' C' F' A2' F' C D2 z4 |]
V:A
G/2 E2 B E A2 E2 B/2 z4 |]
V:S
F/2 F/2 G C2 D/2' E' E' F2' z4 |]
V:A
D F/2 D/2 F2 C G B/2 F/2 z4 |]
[Sample 0] Melody 1 ranking (out of 10)? 1
Melody 2 ranking (out of 10)? 6

[Sample 1] Melody 1 ranking (out of 10)? 2
Melody 2 ranking (out of 10)? 3
```

```
[Sample 2] Melody 1 ranking (out of 10)? 3
Melody 2 ranking (out of 10)? 4

average fitness of population 4 = 7.875


Generation 4 population ...

1     #<MUSIC #x1ABB1CFD>  17/2
2     #<MUSIC #x1ABB1D2D>  7
3     #<MUSIC #x1ABB1D5D>  17/2
4     #<MUSIC #x1ABB1D99>  9
5     #<MUSIC #x1ABB20A5>  37/4
6     #<MUSIC #x1ABB2101>  15/2
7     #<MUSIC #x1ABB215D>  15/2
8     #<MUSIC #x1ABB21A1>  15/2
9     #<MUSIC #x1ABB2209>  11/2
10    #<MUSIC #x1ABC5209>  17/2

----------------Selection----------------
X:1
T:Selection
C:Dystopian Tuesday
M:4/4
L:1/4
Q:1/4=120
V:S clef=treble name=Melody1 snm=Melody1
V:A clef=treble name=Melody2 snm=Melody2
%%score [ ( S ) ( A ) ]
K:C
%%MIDI program 0
V:S
D/2 F/2 D C D/2' E' E' F2' z4 |]
V:A
D F B/2 F2 C F2 G/2 z4 |]
V:S
D/2 F/2 D C D/2' E' E' F2' z4 |]
V:A
D F B/2 F2 C F2 G/2 z4 |]
V:S
```

```
D/2 F/2 D C D/2' E' E' z4 |]
V:A
D F B/2 F2 C F2 G/2 z4 |]
[Sample 0] Melody 1 ranking (out of 10)? 1
Melody 2 ranking (out of 10)? 3

[Sample 1] Melody 1 ranking (out of 10)? 5
Melody 2 ranking (out of 10)? 1

[Sample 2] Melody 1 ranking (out of 10)? 3
Melody 2 ranking (out of 10)? 1

average fitness of population 5 = 8.725


Generation 5 population ...

1     #<MUSIC #x1ABB1CE9>  17/2
2     #<MUSIC #x1ABB1D25>  9
3     #<MUSIC #x1ABB1D55>  9
4     #<MUSIC #x1ABB1D91>  17/2
5     #<MUSIC #x1ABB2091>  9
6     #<MUSIC #x1ABB20E1>  17/2
7     #<MUSIC #x1ABB213D>  9
8     #<MUSIC #x1ABB2191>  67/8
9     #<MUSIC #x1ABB21D9>  8
10    #<MUSIC #x1ABDD119>  75/8


------------------MUSIC SAMPLE 10------------------
Melody 1: D/2 F/2 D C D/2' A/2' E' A/2'
Melody 2: D A B/2 F2 A F2 E2

Total Rank: 75/8
Melody1 Rank: 9/2
Melody2 Rank: 39/8
-------------------------------------------------

---------------Individual 10----------------
X:1
T:Individual 10
```

```
C:Dystopian Tuesday
M:4/4
L:1/4
Q:1/4=120
V:S clef=treble name=S
V:A clef=treble name=A
%%score [ ( S ) ( A ) ]
K:C
V:S
%%MIDI program 0
D/2 F/2 D C D/2' A/2' E' A/2'
V:A
D A B/2 F2 A F2 E2
NIL
```

Code

```
; Method to generate the next generation based on the current population
( defmethod next-generation ( ( cp population ) &aux np )
    ( setf np ( empty-population cp ) )
    ( perform-copies cp np )
    ( perform-crossovers cp np )
    np
)


; Constant for the total number of generations
( defconstant *nr-generations* 5 )


; The genetic algorithm
( defmethod ga ( &aux p )
    ( setf p ( initial-population ) )
    ( display p )
    ( dotimes ( i *nr-generations* )
        ;( if ( = (mod i 10 ) 0 ) ( setf *interactive-generation* t ) )
        ( interactive-selection ( select-individuals p ))
        ( setf p ( next-generation p ) )
```

```
        ( check-average p )
        ( display p )
        ;( setf *interactive-generation* nil )
    )
    ( terpri )
    ; The most fit music sample is displayed at the end
    ( display-music-sample ( most-fit-music-sample (
population-individuals p ) ) )
    ;( display p )

)

; Method to check the eaverage of a population
( defmethod check-average ( ( p population ) )
    ( format t "average fitness of population ~A = ~A~%"
        ( population-generation p )
        ( average p )
    )
)
```

Some Thoughts

Based on the miniscule testing I did with this task, I decided to add a separate task for fine-tuning the parameters of the genetic algorithm. There is an intricate balance needed between variation and user-interaction. Too much variation would eliminate the user's ranking, but not enough variation adds to user fatigue and results in a stagnant population. I am hoping that the next two tasks will fix the issue of copies dominating the population. I will add some longer compositions in the future; this task was just a "proof-of-work" for the GA.