## **Task 15 - Generate Collection of Compositions**

In this task, I experimented with the different parameters of the genetic algorithm to see what effects it has on the sample generation. Unfortunately, I did not do a lot of sampling due to time constraints and fatigue. The experiments below took a few hours to process (due to messing with code and the time it takes to listen to samples). I discuss my interpretation of these results in more detail in the following task.

## **The Experiments**

Album Name (as suggested by ChatGPT): Synthetic Soulscape

Song Name (as suggested by ChatGPT)	Test	Parameters	Most Fit Sample	Notes
Artificial Awakening	User-rank every other generation with average population size	<pre>*beat-total* = 24 *nr-generations* = 10 *user-interaction -g* = 2 *pc-m* = 50 *pc-c* = 40 *pc-x* = 60 *population-size* = 20 *selection-size* = 3</pre>	M USIC SAMPLE 3 Melody 1: B F B C2 G/2 F2 F E2 B2 C2 D2 G/2 F2 B F2 C2 B A/2 D D Melody 2: C D/2 B/2 B C C2 D A/2 B E2 A F/2 B2 E/2 E/2 F/2 F/2 E B/2 G2 Total Rank: 9 Melody1 Rank: 5 Melody2 Rank: 4 	<ul> <li>Not really my favorite of the samples</li> <li>Didn't have issues with the sample convergin g too quickly</li> <li>But the problem is: how do I make it converge at the end?</li> <li>Time Elapsed: 14:05</li> <li>Would this be</li> </ul>

				easier with shorter melodies?
Neon Nightmares	User-rank every generation with average population size	<pre>*beat-total* = 24 *nr-generations* = 10 *user-interaction -g* = 1 *pc-m* = 50 *pc-c* = 40 *pc-x* = 60 *population-size* = 20 *selection-size* = 3</pre>	M USIC SAMPLE 2 Melody 1: G/2' C2' E' D2' C' A' G/2' B' G/2' A2' B' G2' E' F2' E' B/2' D2' A' C' E' Melody 2: F4 G4 E4 E4 F4 E4 C4 D4 E2 G B2 E F G E4 F2 C2 E2 E4 E2 Total Rank: 9 Melody1 Rank: 5 Melody2 Rank: 4	<ul> <li>Mutation got out of hand on this one</li> <li>Same issue with needing to find a better way to converge at the end</li> <li>The crossovers 'ranking should be based off of some parameter related to the parents, but I do not want things to converge too early</li> <li>Changing the double-cr ossover to new weighting after this test :)</li> <li>Time Elapsed: 22 minutes</li> </ul>
Machine	User-rank	*beat-total* = 24	M	Converged very

Messiah	every other	*nr-generations*	USIC SAMPLE	quickly.
	generation with small population size	<pre>= 10 *user-interaction -g* = 2 *pc-m* = 50 *pc-c* = 40 *pc-x* = 60 *population-size* = 10 *selection-size* = 10</pre>	Melody 1: B/2' D/2' F2' F' C' B' A2' C' E2' A' F' G' A2' D/2' D' E/2' B/2' D2' A/2' Melody 2: G G2 F2 D F2 C2 G/2 B/2 A C2 G/2 F C2 B/2 B2 A/2 A2 Total Rank: 15 Melody1 Rank: 7 Melody2 Rank: 8	I honestly put in random rankings for the rest of the generation because it converged so quickly. Theory: It's probably because of the small population size
Chaos Engine	User-rank every generation with small population size	<pre>*beat-total* = 24 *nr-generations* = 10 *user-interaction -g* = 0 *pc-m* = 50 *pc-c* = 40 *pc-x* = 60 *population-size* = 5 *selection-size* = 5</pre>	M USIC SAMPLE 1 Melody 1: B B C/2 B A F E2 G/2 D/2 G/2 F A/2 A/2 G B2 C' G2' A' E2' E2' F' C/2' A/2' F2' Melody 2: E2' B' B2' D2' F2' D' B/2' C/2' A/2' E/2' F/2' G' F/2' E' G/2' B/2' G/2' B' D' F/2' F' E/2' G2' A/2' B/2' E/2' Total Rank: 10 Melody1 Rank: 5 Melody2 Rank: 5	Convergence happened again, albeit not as quickly as the previous population. It seems that genetic algorithm needs a sizable population to prevent from converging to early I threw in random rankings again because it converged so soon.
Rust in the System	No copies with small population size	<pre>*beat-total* = 24 *nr-generations* = 10 *user-interaction -g* = 0</pre>	M USIC SAMPLE 1 Melody 1: A' C2' E2' G E/2 C/2 F G/2 A/2 G/2 A	Melodies converged quickly again so I put in random numbers until the test finished.

		<pre>*pc-m* = 50 *pc-c* = 0 *pc-x* = 100 *population-size* = 5 *selection-size* = 5</pre>	A/2 F B F E G2 B/2 D B D F/2 A/2 F2 C A E2 B Melody 2: F/2' D2' F2' E/2' E2' B' D2' C/2' A/2' F/2' F' A2' G2' E/2' B/2' F2' B/2' E/2' Total Rank: 2 Melody1 Rank: 1 Melody2 Rank: 1	
Synthetic Siren	Dummy test	<pre>*beat-total* = 24 *nr-generations* = 100 *user-interaction -g* = 100 *pc-m* = 50 *pc-c* = 40 *pc-x* = 60 *population-size* = 20 *selection-size* = 3</pre>	M USIC SAMPLE 1 Melody 1: A2 A G/2 E2 B G/2 G/2 D/2 F2 C/2 D2 B G2 B/2 G G/2 D/2 B2 G2 E2 A2 A2 D/2 Melody 2: A' G4' B' C' E2' A4' F' A4' E2' C' A4' F' A4' E2' C' A4' F' A' A4' A4' D' G' C2' E4' E' F2' E4' F' A' Total Rank: 17 Melody1 Rank: 8 Melody2 Rank: 9	Not sure if there is much a difference between the user guided samples and the random samples besides the mutation going out of control and making the second melody much longer than the first one

Electric Eden	Test 1 repeated with smaller beat total	<pre>*beat-total* = 8 *nr-generations* = 10 *user-interaction -g* = 2 *pc-m* = 50 *pc-c* = 40 *pc-x* = 60 *population-size* = 20 *selection-size* = 3</pre>	-I accidentally closed CLisp and lost the sample-	The aforementioned issue of convergence is really an issue of a lack of noticeable variation. With a smaller beat-total, the mutation feels much more prominent and makes the sample feel new. In large samples, changing a single note is rather unnoticeable.
---------------	--	---	---	--