



---

# Interactive Evolution for Melody Generation

— Noemi Canovi, Roberto Mazzaro —  
Michele Presutto, Olga Zaghen

---





# INTRODUCTION

## Interactive Evolution for Melody Generation

*Aim:*

Generating melodic and rhythmic lines

*How:*

Using evolutionary algorithms and interactive fitness

# Introduction

## GA

- A Genetic Algorithm (GA) evolves the population of tracks
- The fitness leads the evolution process



# Introduction

## Subjective Fitness

PRO:

- no complex metric
- the algorithm will suit the user taste

CON:

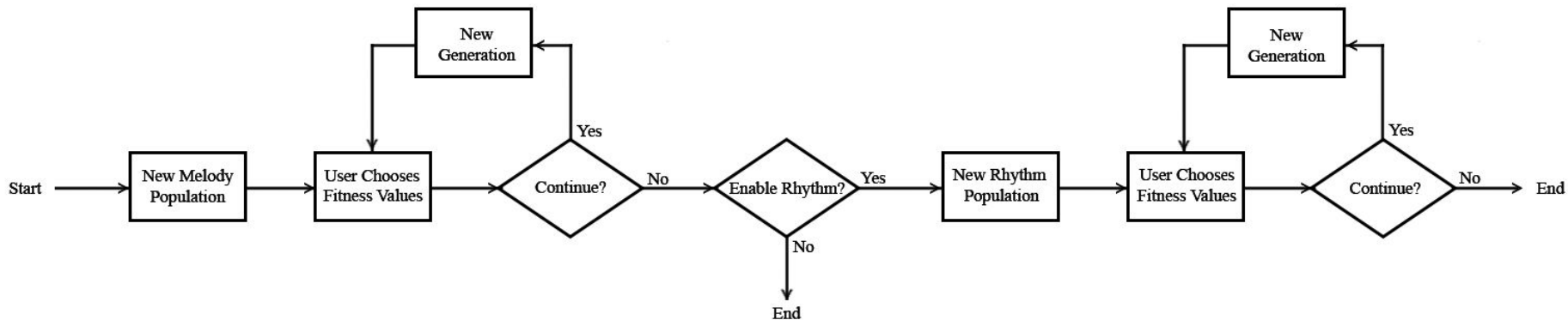
- user's fatigue

Track:		Evaluation:	Default score = 1
1	Press to listen	1 2 3 4 5 6 7 8 9 10	Current score: 2.
2	Press to listen	1 2 3 4 5 6 7 8 9 10	Current score: 7.
3	Press to listen	1 2 3 4 5 6 7 8 9 10	Current score: 6.
4	Press to listen	1 2 3 4 5 6 7 8 9 10	Current score: 5.
5	Press to listen	1 2 3 4 5 6 7 8 9 10	Current score: 3.
6	Press to listen	1 2 3 4 5 6 7 8 9 10	Current score: 7.
7	Press to listen	1 2 3 4 5 6 7 8 9 10	Current score: 3.

*Example of fitnesses chosen by the user*

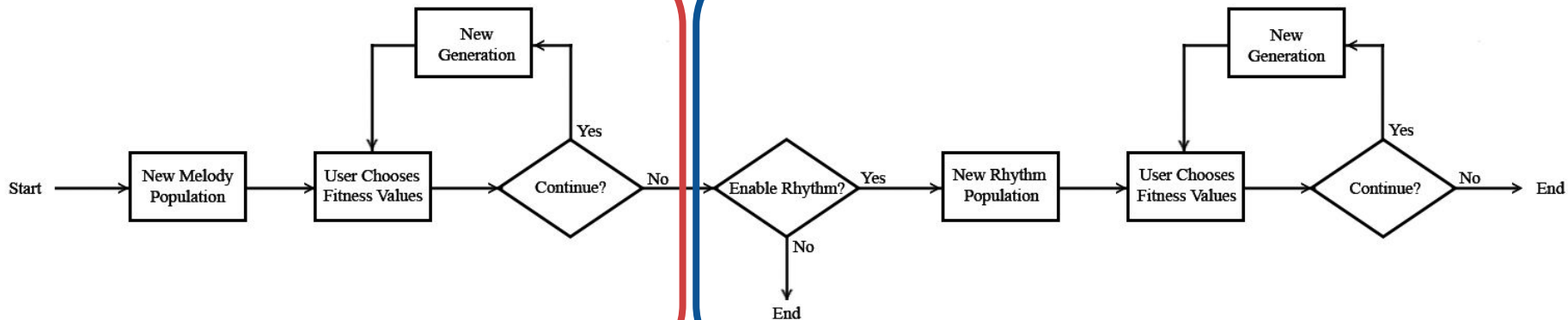
# Introduction

## Main Steps



# Introduction

## Main Steps



*Melody Evolution*

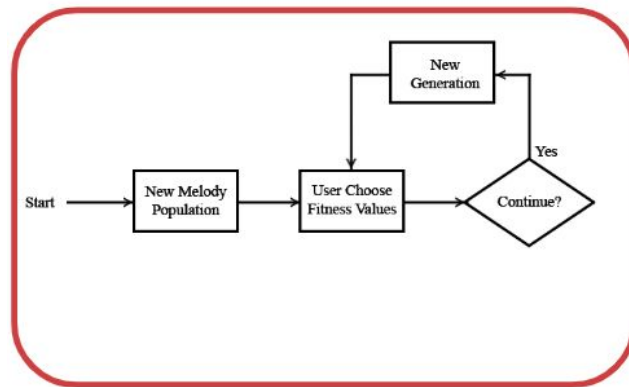
*Rhythm Evolution*

## Introduction

# Melody Evolution: Initial Frame

First interface where the user can choose:

- number of octaves
- population size



Select options:

Number of Octaves (default value = 1):

1

2

3

2 octaves

Number of Melodies (default value = 10):

7

7 melodies

Start melody creation

*Example of values chosen by the user*

## Introduction

# Melody Evolution: Generation

The **melodies** are

- **created**
- **converted** to listenable files (.wav)
- **presented** to the user

Track:		Evaluation:	Default score = 1
1	Press to listen	1 2 3 4 5 6 7 8 9 10	
2	Press to listen	1 2 3 4 5 6 7 8 9 10	
3	Press to listen	1 2 3 4 5 6 7 8 9 10	
4	Press to listen	1 2 3 4 5 6 7 8 9 10	
5	Press to listen	1 2 3 4 5 6 7 8 9 10	
6	Press to listen	1 2 3 4 5 6 7 8 9 10	
7	Press to listen	1 2 3 4 5 6 7 8 9 10	



## Introduction

# Melody Evolution: Evaluation

The user **listen** and **rates**  
the tracks

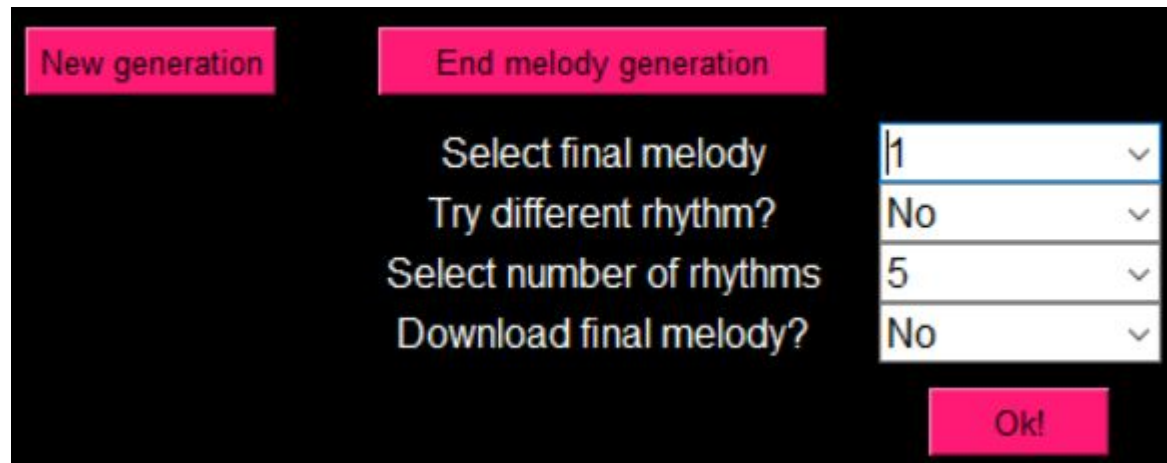
Track:		Evaluation:		Default score = 1
1	Press to listen		1 2 3 4 5 6 7 8 9 10	Current score: 2.
2	Press to listen		1 2 3 4 5 6 7 8 9 10	Current score: 7.
3	Press to listen		1 2 3 4 5 6 7 8 9 10	Current score: 6.
4	Press to listen		1 2 3 4 5 6 7 8 9 10	Current score: 5.
5	Press to listen		1 2 3 4 5 6 7 8 9 10	Current score: 3.
6	Press to listen		1 2 3 4 5 6 7 8 9 10	Current score: 7.
7	Press to listen		1 2 3 4 5 6 7 8 9 10	Current score: 3.

## Introduction

# Melody Evolution: Options

Once **satisfied** with the melody, the user can choose:

- final melody
- to evolve rhythm
- the number of rhythm to evolve at each generation
- to download the final melody



The screenshot shows a dark-themed dialog box titled "End melody generation" in a pink header bar. It contains four options, each with a corresponding pink button to its right:

- New generation** (pink button)
- Select final melody** (pink button)
- Try different rhythm?** (pink button)
- Select number of rhythms** (pink button)
- Download final melody?** (pink button)

Below the buttons, there is a table with four rows, each corresponding to one of the options above. Each row has a text input field and a dropdown arrow:

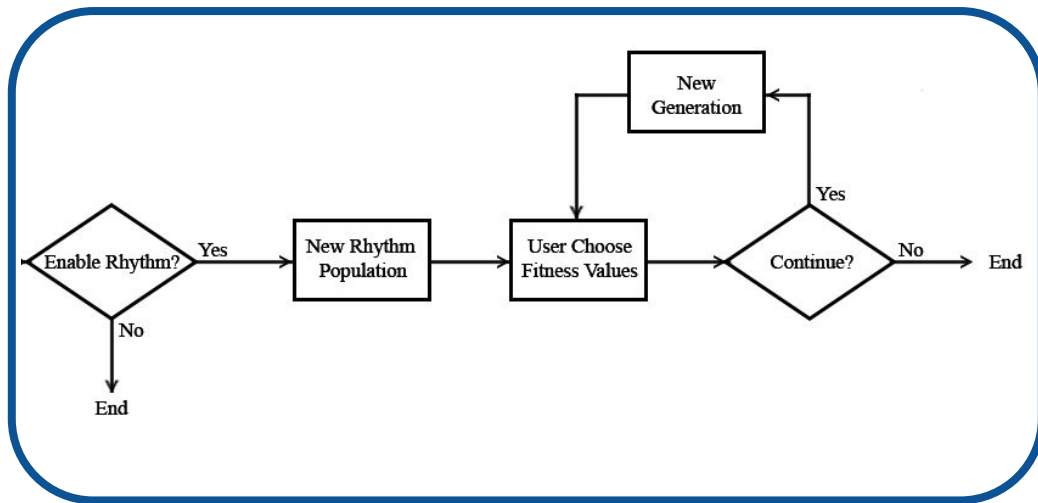
Select final melody	1	▼
Try different rhythm?	No	▼
Select number of rhythms	5	▼
Download final melody?	No	▼

At the bottom right of the dialog box is a pink button labeled "Ok!".

# Introduction

## Rhythm Evolution

The **same steps** are applied here  
for the **rhythm evolution**!



The only difference is the  
**options** that pop up when the  
user is satisfied with the track

A screenshot of a user interface with a black background. At the top, there are two pink buttons: 'New generation' on the left and 'End rhythm generation' on the right. Below these buttons, the text 'Select final rhythm' is followed by a dropdown menu showing the number '1'. Below that, the text 'Download final rhythm?' is followed by a dropdown menu showing the word 'No'. At the bottom right, there is a pink button labeled 'Ok!'. A thin grey arrow points from the text 'options that pop up when the user is satisfied with the track' to the 'End rhythm generation' button.



# ENCODING

A main **issue** in GA is how to **encode** problem specific **informations into genes**

- Individual encoding
- Note encoding - Pitch
- Note encoding - Rhythm
- Obtain Phenotype

# Individual encoding

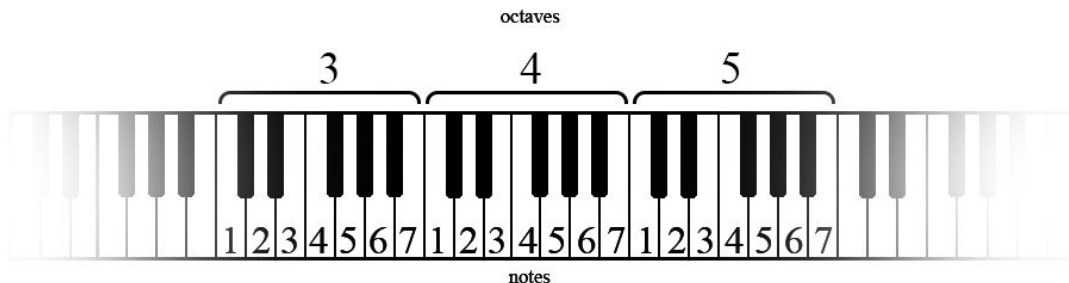
- Individual is a **list of notes**
- Length: **8**
- Implemented in the class **Notelist**

Individual 1							
Measure 1				Measure 2			
V = 6	V = 1	V = 4	V = 3	V = 2	V = 4	V = 1	V = 6
O = 4	O = 4	O = 4	O = 4	O = 4	O = 4	O = 4	O = 4
R = 2	R = 2	R = 4	R = 3	R = 2	R = 2	R = 3	R = 2

*Example of individual*




# Note encoding - Pitch

- Note described from **value**, **octave** and **rhythm**
- Implemented in the class **Note**
- Work with max. **3 octaves**
- **No alterations** to the note value
- Octave value from a **Gaussian centered in 4**
- Note value belongs to  $\{1, \dots, 7\}$  generated with **Zipf's law**



# Note encoding - Rhythm

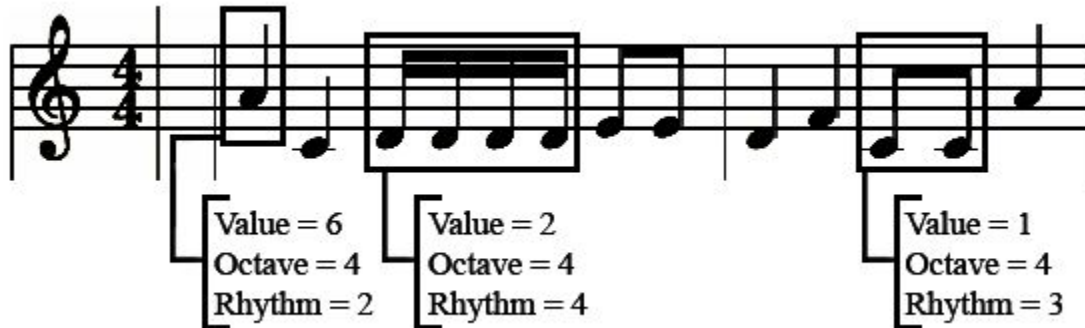
- Rhythm is **simplified** as a **repetition** of the **same note** with **different durations**
- Available note durations: 1/4, 1/8, 1/16
- Encoded as the **power of  $\frac{1}{2}$**
- The value is always 2 in the melody evolution
- In the rhythm evolution generated from right side of a gaussian centered in 2

2	3	4
		

# Individual and Note encoding

An individual has this form:

Individual 1							
Measure 1				Measure 2			
V = 6	V = 1	V = 4	V = 3	V = 2	V = 4	V = 1	V = 6
O = 4	O = 4	O = 4	O = 4	O = 4	O = 4	O = 4	O = 4
R = 2	R = 2	R = 4	R = 3	R = 2	R = 2	R = 3	R = 2





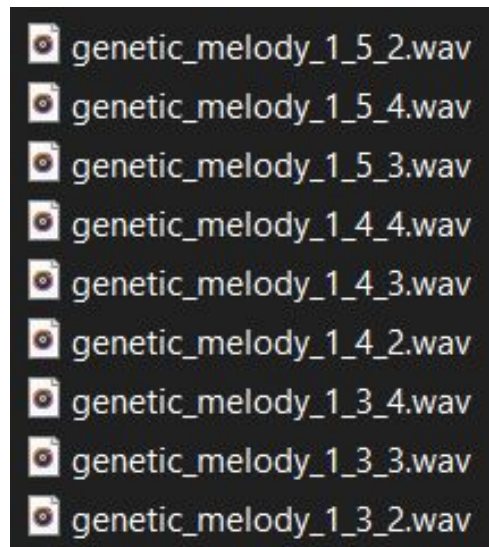
# From Genotype to Phenotype

## Phenotype:

- User have to **listen** to the generated individual
- Use Wave to **append** Note **sounds**
- Saved as indiv\_n

## Dataset:

- **63 audio** files (7x3x3)
- Named as genetic\_melody\_V\_O\_R
- Sampled from an electric piano
- Beat 104 bpm



*All different files for C note*



# GENETIC ALGORITHM

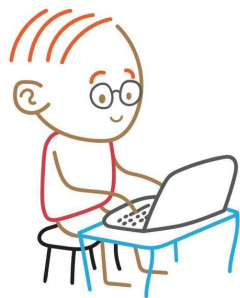
1. Subjectivity of the fitness
2. GA: algorithmic choices
  - Population and generations
  - Parent selection
  - Crossover
  - Mutation

# Subjective fitness



Centrality of the **user interaction**  
in the evolutionary computation

# Subjective fitness

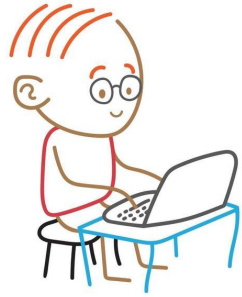


Centrality of the **user interaction**  
in the evolutionary computation

**Fitness** of a melody  
=  
rate (**from 1 to 10**) of the user

subject-dependent

# Subjective fitness



Centrality of the **user interaction**  
in the evolutionary computation

**Fitness** of a melody  
=

rate (**from 1 to 10**) of the user

subject-dependent

Influence of **human fatigue**  
on the parameters of the GA

GA: algorithmic choices

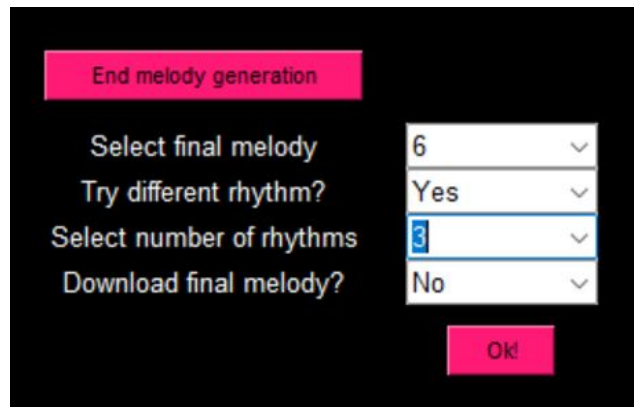
# Population and generations

- The **number of generations** and the **population size** are lower than classic GA settings (**human fatigue**)

## GA: algorithmic choices

# Population and generations

- The **number of generations** and the **population size** are lower than classic GA settings (**human fatigue**)



A settings menu with a black background and pink text and buttons. It includes options for ending a generation, selecting a final melody, trying a different rhythm, selecting the number of rhythms, and downloading the final melody. Each option has a corresponding dropdown menu.

End melody generation	
Select final melody	6
Try different rhythm?	Yes
Select number of rhythms	3
Download final melody?	No
OK	

- The **melody** population size can be set to a value from 1 to 10
- The **rhythm** population size can be set to a value from 1 to 5

Select options:

Number of Octaves (default value = 1):

1

2

3

2 octaves

Number of Melodies (default value = 10):

7

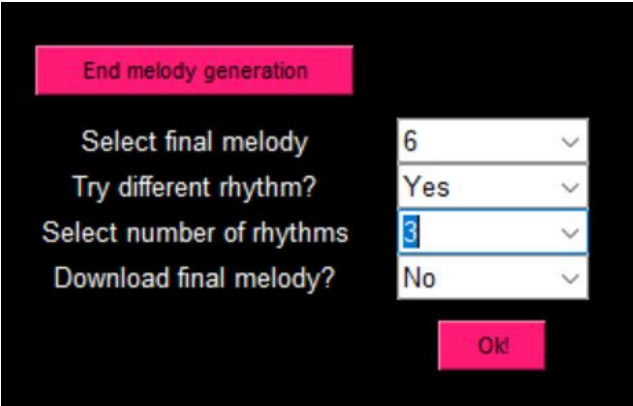
7 melodies

Start melody creation

## GA: algorithmic choices

# Population and generations

- The **number of generations** and the **population size** are lower than classic GA settings (**human fatigue**)



A dialog box with a black background and pink buttons. It contains four rows of settings, each with a label and a dropdown menu. The labels are 'End melody generation', 'Select final melody', 'Try different rhythm?', and 'Select number of rhythms'. The dropdown values are '6', 'Yes', '3', and 'No' respectively. An 'Ok!' button is at the bottom right.

End melody generation	
Select final melody	6
Try different rhythm?	Yes
Select number of rhythms	3
Download final melody?	No
Ok!	

- The number of generations depends on when the **user** ends the algorithm

Select options:

Number of Octaves (default value = 1):

1

2

3

2 octaves

Number of Melodies (default value = 10):

7

7 melodies

Start melody creation



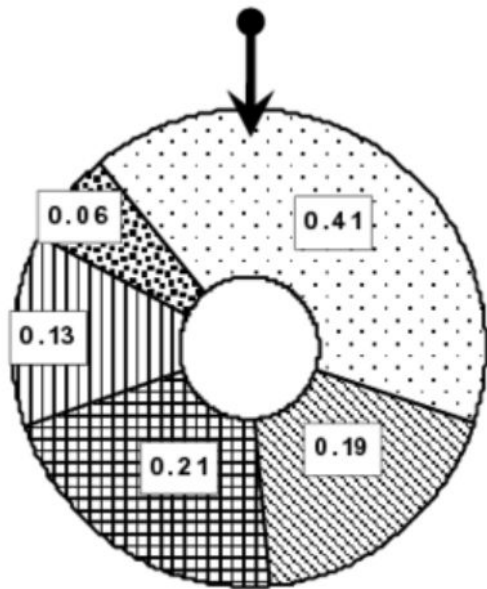
GA: algorithmic choices

## Parent selection

**Roulette wheel** prediction formula

**Fitness-proportionate** parent selection

Promote **exploitation**



$$p_i = \frac{f_i}{\sum_{j=1}^N f_j}$$

GA: algorithmic choices

# Survivor selection

- ★ **Replace** the old population by offspring
- ★ Maintain some **elite** individuals



GA: algorithmic choices

# Survivor selection

★ **Replace** the old population by offspring

★ Maintain some **elite** individuals



**Their quantity depends  
on the population size**



# GA: algorithmic choices

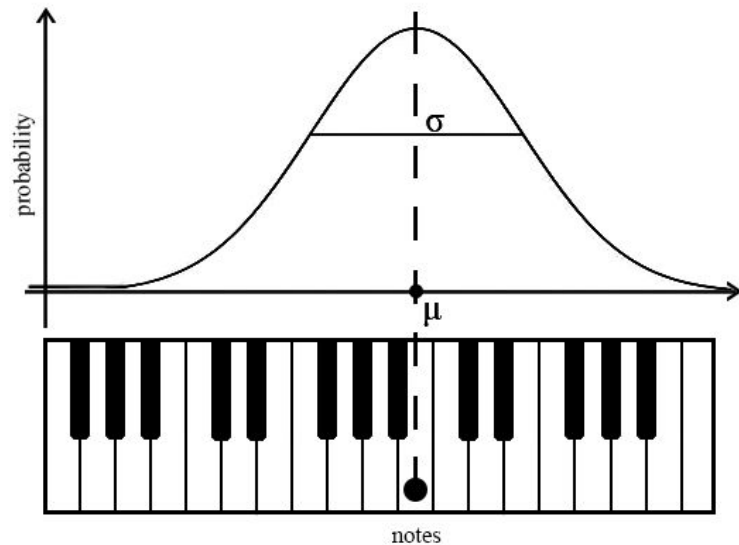
## Mutation

### Melody:

- **Gaussian** centered in the actual **note position**
- Generate a new different position
- Recover new note value and octave

### Rhythm:

- New random value from the **half** Gaussian distribution centered in **2**



*Representation of melody's mutation*

# GA: algorithmic choices

## Crossover

### Melody:

- One point crossover
- Between 2 Notelist

### Rhythm:

- One point crossover
- Only rhythm values

Individual 1							
Measure 1				Measure 2			
V = 6	V = 1	V = 4	V = 3	V = 2	V = 4	V = 1	V = 6
O = 4	O = 4	O = 4	O = 4	O = 4	O = 4	O = 4	O = 4
R = 2	R = 2	R = 2	R = 2	R = 2	R = 2	R = 2	R = 2

Individual 2							
Measure 1				Measure 2			
V = 1	V = 2	V = 3	V = 1	V = 1	V = 2	V = 3	V = 1
O = 4	O = 4	O = 4	O = 4	O = 4	O = 4	O = 4	O = 4
R = 2	R = 2	R = 2	R = 2	R = 2	R = 2	R = 2	R = 2

Generated Individual							
Measure 1				Measure 2			
V = 1	V = 2	V = 3	V = 1	V = 1	V = 4	V = 1	V = 6
O = 4	O = 4	O = 4	O = 4	O = 4	O = 4	O = 4	O = 4
R = 2	R = 2	R = 2	R = 2	R = 2	R = 2	R = 2	R = 2

*Melody Crossover*

Generated Individual							
Measure 1				Measure 2			
V = 6	V = 1	V = 4	V = 3	V = 2	V = 4	V = 1	V = 6
O = 4	O = 4	O = 4	O = 4	O = 4	O = 4	O = 4	O = 4
R = 2	R = 2	R = 4	R = 2	R = 3	R = 2	R = 4	R = 2

*Rhythm Crossover*



# STATISTICS

We carried out an analysis to  
**evaluate** the effectiveness  
of our system

## Statistics

# Methodology

- We stored the **average** score and **maximum** score in each generation, for **40** different executions



28151915 - Blocco note

File	Modifica	Visualizza
2.5	4	
3.0	4	
3.25	4	
4.0	5	
3.0	5	
4.75	6	
5.25	7	
2.5	7	

## Statistics

# Methodology

- We stored the **average** score and **maximum** score in each generation, for **40** different executions
- We performed:
  1. Statistics on **number of generations**
  2. Analysis of the **fitness trend** by averaging results over all executions



28151915 - Blocco note

File	Modifica	Visualizza
2.5	4	
3.0	4	
3.25	4	
4.0	5	
3.0	5	
4.75	6	
5.25	7	
2.5	7	



## Statistics

# Number of generations

The average number of iterations in evolution resulted to be



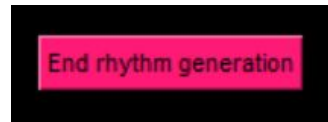
**4.96**

for the melody



**1.67**

for the rhythm



**total** 10-15 mins of runtime

## Statistics

# Number of generations

The average number of iterations in evolution resulted to be



4.96

for the melody



1.67

for the rhythm



**total** 10-15 mins of runtime

not subject to proper evolution

## Statistics

# Fitness trend for melody evolution

absolute fitness value

fitness gain

average score	3.75	5.10	1.46
max score	5.67	7.30	1.76
	first gen	last gen	last - first

## Statistics

# Fitness trend for melody evolution

	absolute fitness value		fitness gain
average score	3.75	5.10	1.46
max score	5.67	7.30	1.76
	first gen	last gen	last - first



Tendency to a substantial **positive gain**

## Statistics

# Fitness trend for melody evolution

	absolute fitness value		fitness gain
average score	3.75	5.10	1.46
max score	5.67	7.30	1.76
	first gen	last gen	last - first



Tendency to a substantial **positive gain**



The algorithm well adapts to the **subjective taste** of the human user in the melody evolution!



# **DIFFICULTIES & IMPROVEMENTS**

## *Difficulties*

Issues we encountered and how we resolved them

## *Improvements*

Possible future works

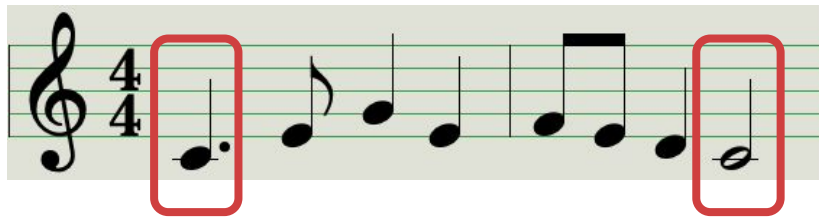
# Difficulties



- **Relative** fitness
- **Fitness bottleneck**
- Implementation of the **rhythm**
- **Hyperparameters** difficult to set
- High influence of the **first generation**

# Improvements

- Assumptions **relaxation** for note and rhythm encoding
- Adding more **constraints**
- Defining an **objective fitness function**



*Melody that cannot be generated using the actual encoding*

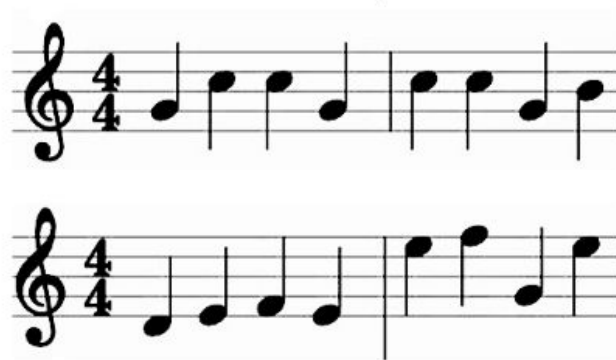




# CONCLUSIONS

- ★ Nice melody can be generated:
  - with few constraints
  - in a few number of generation
- ★ We collected the most interesting results

# Interesting creations



Example of two melodies produced by the GA.

They have been generated in the same evolutionary process and they are **objectively musically coherent**, in addition to complementing each other well!



**THANKS FOR YOUR ATTENTION!**



# Difficulties

**subjectivity and relative fitness:** the fitness is different from person to person, and is biased by the overall score of the other melody in the context.

**fitness bottleneck:** due to the active participation of the user in the listening of each individual and the setting of a score to each melody, it is likely to get tired after some iterations.

**influence of the first generation:** if the first generation is good then the number of generations to obtain a satisfactory score will be low. We tried to solve this issue by using zipf's law.

# Difficulties

**implementation of the rhythm:** implementing the rhythm structure maintaining a fixed length and duration of the melodies.

**Hyperparameters difficult to sets:** finding the best setting for the hyperparameters of the GA without burdening the user.

**High influence of the first generation:** the convergence of the GA is faster with at least a good individual in the initial populations while it stagnate without



# IMPROVEMENTS

Possible improvements could be:

- Assumption relaxation for note and rhythm encoding
- Adding more constraints
- Defining an objective fitness function



# CONCLUSION

we can say that the interactive evolution computation system that we implemented is capable of producing nice melodies in a restricted number of generations

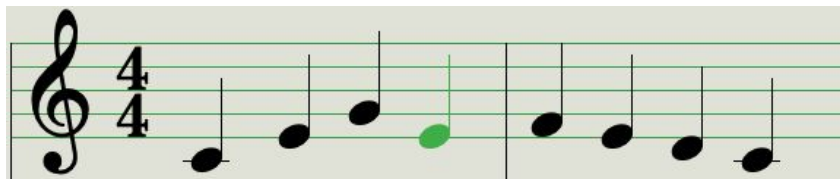
# Statistics

## Improvements

### Assumption and relaxation for rhythm encoding:

the actual encoding of the rhythm consist of splitting the same note by maintaining its total duration.

maybe a better encoding could be changing just the duration of the note and doing a control on the entire melody length



melody that can generate  
using actual encoding



melody that can not generate  
using actual encoding



# Interesting creations



Example of two melodies produced by the GA.

They have been generated in the same evolutionary process and they are **objectively musically coherent**, in addition to complementing each other well!

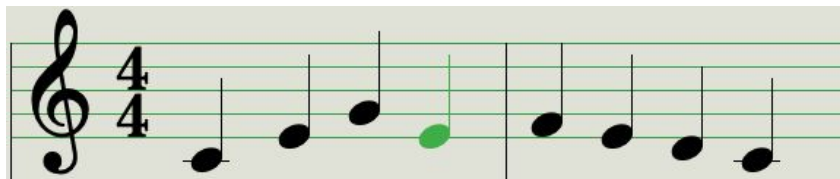
## Improvements

# Rhythm encoding

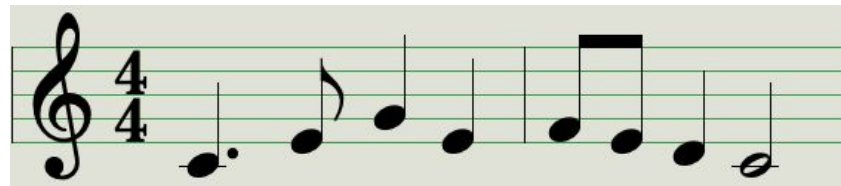
repetition of the same note maintaining its total duration



changing just the duration of the note and doing a control on the entire melody length

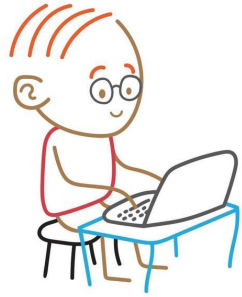


melody that can generate  
using actual encoding



melody that can not generate  
using actual encoding

# Subjective fitness



Centrality of the **user interaction**  
in the evolutionary computation

**Fitness** of a melody

=

rate (**from 1 to 10**) of the user

subject-dependent

Influence of **human fatigue**  
on the parameters of the GA

# Subjective fitness



Centrality of the **user interaction**  
in the evolutionary computation

**Fitness** of a melody

=

rate (**from 1 to 10**) of the user

## Statistics

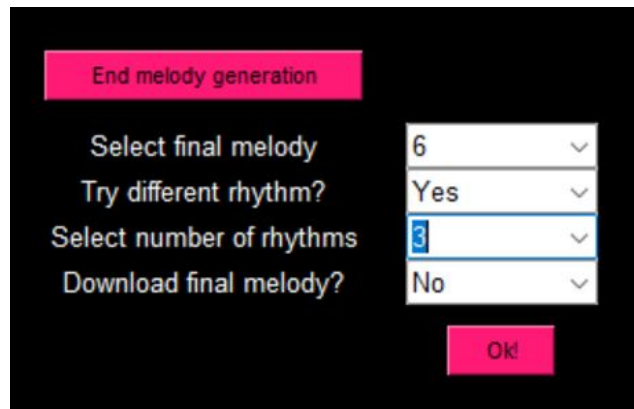
### Results : fitness trend

	absolute fitness value		fitness gain	
average score	3.75	5.10	1.46	0.35
max score	5.67	7.30	1.76	0.42
	first gen	last gen	raw = last - first	mean = last - first <hr/> # generations

## GA: algorithmic choices

# Population and generations

- The **number of generations** and the **population size** are lower than classic GA settings (**human fatigue**)
- The **melody** population size can be set to a value from 1 to 10, **a priori**
- The **rhythm** population size can be set to a value from 1 to 5, **a priori**
- The number of generations depends on when the **user** ends the algorithm, **a posteriori**



A dialog box titled "End melody generation" with a pink header bar. It contains four labels with corresponding dropdown menus: "Select final melody" (value 6), "Try different rhythm?" (value Yes), "Select number of rhythms" (value 3), and "Download final melody?" (value No). All dropdown menus have a downward arrow icon. At the bottom right is a pink "OK" button.

Label	Value
Select final melody	6
Try different rhythm?	Yes
Select number of rhythms	3
Download final melody?	No

Select options:

Number of Octaves (default value = 1):

1

2

3

2 octaves

Number of Melodies (default value = 10):

7

7 melodies

Start melody creation