Sessie 6

ASSIGNMENT - Rhythm generation strategies

TIMEBOX!
2 hours - max.

1. Form duos / trios
2. Choose a rhythm generation strategy
   See next slide (# 3)!
3. Research:
   a. the strategy in an overall manner
   b. strategy's algorithm(s)
   c. pros and cons of the strategy
   d. optional: the strategy in pseudo code
4. Prepare 1 to 3 slides to present during the next session
   ADD THESE HERE, AFTER SLIDE 3!
   You are free to use a blueprint slide (not yet in English).

ASSIGNMENT - Rhythm generation strategies

TIMEBOX!
2 hours - max.

- ~~Probability based on position in measure~~
- ~~Euclidean rhythms~~
- ~~Midpoint displacement~~
- 1st order markov chain
- Sets of 2s and 3s as building blocks → *Roman James & Pier*
- Sequence variations on given configuration sets
- A totally different strategy, none of the above
  e.g. using the weather or an image as input

Sets of 2 and 3’s as building blocks

Pier & Roman-James

the strategy

input = hoeveelheid 8ste noten

verdeeld input in 2’en en 3’en in een lijst

shuffelt lijst

output = lijst

Pros and cons

Pros:

- makkelijk om in te voeren
- werkt goed met oneven maatsoorten
- creëert ritmes die vaak voorkomen in volksmuziek

Cons:

- maakt alleen het skelet van het ritme
- alleen 2’s en 3’s

```python
import random

num = 8 # natural number between 2 and ∞

print("input: " + str(num))

array = []

while True:
    if num == 6:
        array.append(3)
        array.append(3)
        break
    elif num == 3:
        array.append(3)
        break
    elif num == 2:
        array.append(2)
        break
    else:
        array.append(2)
        num = num-2

random.shuffle(array)

print("output: " + str(array))
```

basis algoritme:

input: 8

output: [2, 3, 2]

ornament:

input: [3, 2, 3]

output: [3, 2, 1.5, 1.5]

```python
import random

array = [3, 2, 3]

rand_int = random.randint(2, 5)

print(rand_int)

last_num = array.pop()

for _ in range(rand_int):
    array.append(last_num/rand_int)

print(array)
```

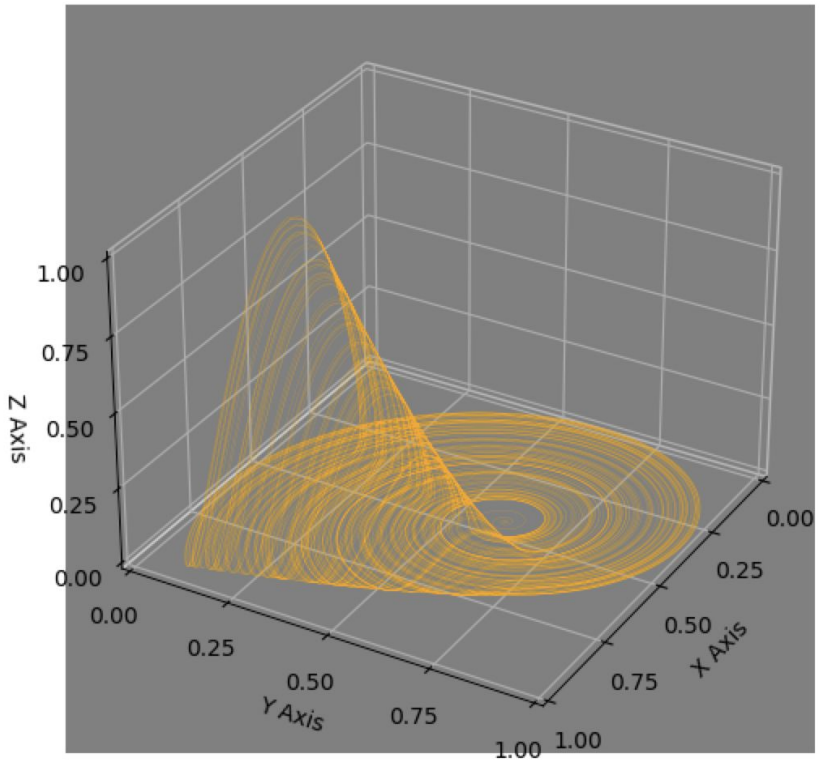
Josef Häusel

Rössler Attractor Sonification

$$\frac{dx}{dt} = -y - z$$

$$\frac{dy}{dt} = x + ay$$

$$\frac{dz}{dt} = b + z(x - c)$$



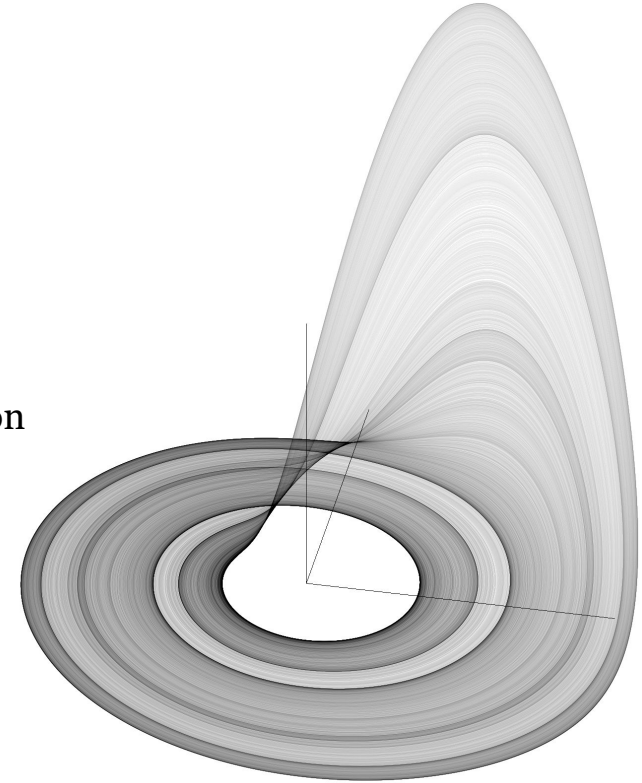
Josef Häusel

Rössler Attractor Sonification

Part I - Calculation
-Program function to run a given function recursively.
-Normalize Data

Part II - Sonification
-Define conditions of what constitutes a note (binary; 0 or 1):
    Method 1: Axis Thresholds
    Method 2: Gradient Thresholds
        -> [0,0,0,1,0,0,01,0,0,1…] for each axis / dimension
-Convert notes to timestamps (Steps-per-Second)
-Multiprocessing Playback
    Assign sample to each axis / dimension

Part III - Plotting
-MatLab



Code:

https://github.com/josefhaeusel/CSD2/tree/main/CSD2a/opdrachten/rossler_attractor

1st-order Markov chain

The strategy
Using a first-order Markov chain to generate a rhythm
Input a rhythm
Calculate probabilities
Generate rhythm using probabilities

Pros
- Some characteristics of input are present
- Can continue indefinitely

Cons
- Does not generate good drum beats
- Has difficulty generating a steady pulse

Pseudocode & demo

```
input = rhythm_input()
events_dict = {}

for event in input:
    events_dict[previous_event].append(event)

current_state = input[0]

while(playing):
    if time_for_16th_note_passed:
        current_state.play()
        current_state.random_next_state()
```

```
low   x-------x-------
mid   ----x-------x---
high  --x---x---x---x-
```