

```
print(f'{Irregular Beat Generator}')  
[Print(i, end='') for i in 'Fabian Dekker']
```

Inhoud

- Essentie generatie strategie
- Rotation
- Reflectie
 - Wat heb ik geleerd
 - Waar liep ik tegenaan
 - Takeaways

Essentie Generatie Strategie

- Euclidean Algorithm

"""sample_settings is een dictionary.
Layer_name is de naam van de laag -> de layers
zijn ook de keys van sample_settings"""

```
for layer_name in sample_settings:
```

```
    # Improves readability
```

```
    num_notes = sample_settings[layer_name]['num_notes']
```

```
    num_pulses = global_settings['num_pulses']
```

```
    rotation_amount = sample_settings[layer_name]['rotation_amt']
```

```
    duration = int(num_pulses / num_notes)
```

```
    remainder = num_pulses - (num_notes * duration)
```

```
    duration_sequence = [duration] * num_notes
```

```
    # Distributes remainder amongst values stored in duration_sequence
```

```
    for j in range(remainder):
```

```
        duration_sequence[j] += 1
```

Essentie Generatie Strategie

- Algorithm generates duration sequence
 - Rotation
 - Durations to indexes
 - Rotate indexes
 - Calculate Rest

```
""" duration_list to index_list """
```

```
for index, i in enumerate(input_list):  
    for j in range(i):  
        index_list.append(index)
```

```
""" [2,2,1,1,2] -> [0,0,1,1,3,4,5,5] """
```

```
"""Rotate index_list"""
```

```
def rotate_indexes(inp_list):  
    if amount > 0:          # rotate_right  
        return inp_list[-wrapped_amt:] + inp_list[:-wrapped_amt]  
    elif amount < 0:        # rotate_left  
        return inp_list[wrapped_amt:] + inp_list[:wrapped_amt]  
    else:  
        return inp_list
```

```
"""[0,0,1,1,3,4,5,5] -> rotate 1 (to right) -> [5,0,0,1,1,2,3,5]"""
```

```
"""index_list to duration and calculate rest"""
```

- Kijk welke duratie bij de eerste index hoort
 - In dit geval 2
 - `Output_list.append(2)`
- Kijk naar de eerste twee indexes
 - `False` if `list[index] != 2`
- Voor elke `False` doe `subtract_value += 1`
 - `Rest_duration = duration_at_index - subtract_value`
- Return (`output_list`, `rest_duration`)

```
"""[5,0,0,1,1,2,3,5] -> ([2,2,1,1,1],1)"""
```


Reflectie

- Wat heb ik geleerd:
 - Ik kan zeer comfortabel werken in python
 - Git workflow
 - Pseudo code nuttig
- Waar liep ik tegenaan:
 - Scope creep
 - Time blindness/time management
- Takeaways:
 - Timeboxen
 - Duidelijk leerdoel stellen en daar op focussen
 - Vaardigheid verkrijgen > perfect eindresultaat van opdracht
 - Hoofdzaak van bijzaak scheiden
 - Sneller tot de essentie komen van wat ik wil vragen/vertellen/presenteren