

Opdracht - Ritmische playback

In deze opdracht maak je een Python script dat een sample een gegeven aantal keren afspeelt in een bepaald ritme.

Verwerk de volgende functionaliteiten in je programma:

- Het script verwacht de volgende input vanuit de terminal:
 - Eén getal dat het aantal keren dat de sample wordt afgespeeld bepaalt, dit noemen we voor nu even “numPlaybackTimes”.
 - Vervolgens worden er *numPlaybackTimes* regels ingevoerd met op elke regel één getal als input. Deze input vormt het gegeven ritme.
 - Tot slot volgt er nog één regel met één getal, dit is de gebruikte BPM voor het afspelen

Een voorbeeld van de te verwachten input:

```
4
1
0.5
1.5
0.5
120
```

- Het script transformeert het gegeven ritme naar een *sequence* van duraties op basis van de gegeven bpm. Het getal 1 staat voor een kwartnoot, een 0.5 is een achtste noot, etcetera. Met de BPM kun je de duur van een kwartnoot berekenen.
- Het script speelt het hele ritme af met één sample naar keuze. De naam hiervan kun je hard-coden of als extra optie aan de gebruiker vragen.

Tips

- Wanneer je bepaalde functionaliteit nodig hebt zoals een “for loop”, dan kun je o.a. de python library reference gebruiken <https://docs.python.org/3.7/library/index.html>. Daarnaast zijn op Lynda.com goede tutorials te vinden, je hebt toegang tot dit platform met je HKU account. (<https://studentportal.hku.nl/studentzaken/random-je-studie/ictdiensten/lynda.com.htm>)
- Let op! Wanneer je uit de terminal iets inleest met de functie *input*, welk data type is dit dan? Kun je dit gebruiken als getal, of moet je het eerst converteren en zo ja, hoe doe je dat in Python?
- Het eerste getal dat je inleest (*numPlaybackTimes*) kun je gebruiken in een loop om de getallen die onderdeel zijn van de ritme-sequence in te lezen. Deze *sequence* van getallen kun je opslaan in een list. Zoek zelf op hoe je een list aanmaakt en hier items aan toevoegt. Test dit uit in een los script.
- Als loop kun je een for loop gebruiken en de functie *range*. Het is goed om dit in een los testscript even uit te proberen. Wat doet *range*? Van waar tot waar loopt *range*? etc.

- Je kunt ervoor kiezen om elke keer handmatig input in te voeren in de terminal, maar je kunt ook wanneer je je script uitvoert direct een bestand meegeven dat de nodige input bevat. Zet je input (bijvoorbeeld het bovenstaande voorbeeld) in een bestand genaamd *input.txt* (je kunt een andere naam en extensie gebruiken).

Wanneer dit bestand zich in dezelfde map bevindt als je script, dan kun je het script als volgt aan roepen vanuit de terminal:

```
python3 myScript.py < input.txt
```

- Gebruik de module *simpleaudio* om een sample mee af te spelen, deze module moet geïnstalleerd worden met *pip* (zie hiervoor ook “hello world” opdracht 5 in de vorige handout, voor python3 gebruik je *pip3*).
- Gebruik de module *time* om je script kort te pauzeren. De *time* module is onderdeel van de Python standard library en hoeft daarom niet apart met *pip* geïnstalleerd te worden.
- Om modules te kunnen gebruiken in je script moet je deze in je script importeren. Zoek online op hoe dit moet.
- Bestudeer de *references* van de modules die je gebruikt; bekijk welke functies onderdeel zijn van de modules en hoe je hiermee kunt werken. Maak losse kleine testscripts waarin je functionaliteiten die je nodig hebt in de opdracht even los uitprobeert voordat je ze in je programma opneemt.
- Loop je vast? Blijf niet eindeloos doormodderen maar bespreek het met anderen.
- Tot slot, denk aan het hanteren van een zinvolle naamgeving en voeg duidelijke comments toe. De algemene beoordelingscriteria voor *CSD jaar 2* zijn te vinden op: <https://csd.hku.nl/csd2/beoordelingscriteria.html>

Optioneel

- polyfone beat: denk eens na over hoe je dat kunt aanpakken en werk dit uit.
- `time.sleep(note_duration)` → hoe nauwkeurig is dit? Wat gebeurt er als elke `time.sleep` een vertraging oploopt? Waar resulteert dat in? Denk na over hoe je dat zou kunnen oplossen en werk dit uit.