Making music with Python

Organised sound

- Music is ‘organised sound’
- A sequencer is a common way to organise sound:
  - https://www.youtube.com/watch?v=3tCctbRcaos

A python sequencer

- We'll build a simple sequencer in Python!

Starting out

- Create a virtual-env ('musicEnv') with Anaconda / Python 3.6
- Open your IDE
- Activate the musicEnv

Install simpleaudio

- Within your virtual-env:

```
pip install --upgrade pip setuptools
pip install simpleaudio
```

- Documentation:
- http://simpleaudio.readthedocs.io/en/latest/installation.html

Clone my git repo, bro

- https://github.com/ruben-yacht/PythonSequencer
- It's a bit large (140 MB) because I included some sound samples to start with
- **We'll write our code in** `coding-music.py`
- `Sequencer.py` **can play a user-provided pattern**
- `RepeatedTimer.py` **handles timing**
- `SoundFile.py` **reads & plays sounds via simple audio**

Playing a sound

hello-bongo.py:

```
from SoundFile import SoundFile
from Sequencer import Sequencer

if __name__ == "__main__":
    sf1 = SoundFile("hello-bongo.wav")
    sq1 = Sequencer(sf1,[1])
    sq1.play(1)
```

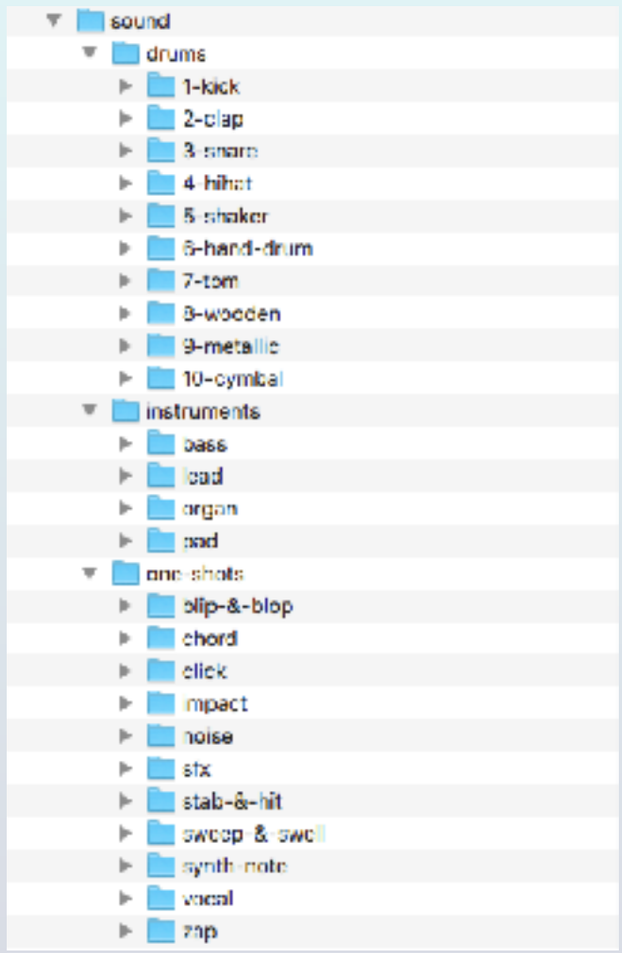
Making a beat

example-basic-drumbeat.py:

```
from SoundFile import SoundFile
from Sequencer import Sequencer

if __name__ == "__main__":
    x = 1
    o = 0

    sf1 = SoundFile("../sound/drums/1-kick/kick-allaboutyou-1.wav")
sq1 = Sequencer(sf1,[x,o,o,o,x,o,o,o])
sq1.play(10)

sf2 = SoundFile("../sound/drums/4-hihat/closedhh-allaboutyou.wav")
sq2 = Sequencer(sf2,[x,o,x,o,x,o,x,o])

sq2.play(10)

sf3 = SoundFile("../sound/drums/3-snare/snare-allaboutyou.wav")
sq3 = Sequencer(sf3,[o,o,o,o,x,o,o,o])
sq3.play(10)
```

Exercise 1

- Explore the sounds
- Make a selection of 3-5 sounds to use
- Copy/note their paths
- Program a simple beat in `coding-music.py`



Polyrhythms

- A sequencer repeats the pattern after the list is finished; so it doesn't have a fixed length like in the video! Try:

```
from SoundFile import SoundFile
from Sequencer import Sequencer

if __name__ == "__main__":
    x = 1
o = 0

sf1 = SoundFile("../sound/drums/1-kick/kick-allaboutyou-1.wav")
sq1 = Sequencer(sf1,[x,o,o])
sq1.play(10)

sf2 = SoundFile("../sound/drums/4-hihat/closedhh-allaboutyou.wav")
sq2 = Sequencer(sf2,[x,o,x,o])

sq2.play(10)

sf3 = SoundFile("../sound/drums/3-snare/snare-allaboutyou.wav")
sq3 = Sequencer(sf3,[o,o,o,o,x,])
sq3.play(10)
```

Exercise 2

- Choose a few new sounds
- Make a beat that only sounds the same once in so many rounds.
- E.g., one sequence of length 3, and one of length 2, only sounds the same once in 3 rounds:

```
[1,0,0,][1,0,0,][1,0,0,]
[1,0,][1,0,][1,0,][1,0,]
```

Editing sounds

- simpleaudio unfortunately doesn't let us change volumes or add effects.

- However, you can edit the .wav files with a free audio editor to craft your masterpiece:

- https://www.audacityteam.org/download/

Exercise 3

- Download & install Audacity
    - Nice Mac alternative: TwistedWave
- Open a sound from your previous polyrhythmic beat
    - ‘read the files directly from the original’
    - select the file
    - effect > amplify > [choose louder or softer] > ok
    - save
- Go back to your IDE and check if the volume sound of the beat changed.

Extra exercises

- Use mathematics to calculate some sequences, so that you don’t know the result in advance.

- Download some files from freesound.org. Make sure it is a 16-bit .wav file, or convert it to that. Use these sounds to make a beat in your own style

Diving deeper

- There seem to be various libraries to make, edit and save wave files, but I didn't have a chance yet to try them all out!

- Can you find a way to:
  - Reverse an audio file
  - Apply an effect to an audio file in pure python - e.g. a reverb, chorus or delay (echo)
  - Generate your own audio file with a 220Hz sine wave