Hi 👋

Dror Ayalon (Dodi)

ABOUT ME

- 2004 -> 2007
  Intelligence at IDF

- 2008 -> 2011
  BA in Communication and Human-Computer Interactions

- 2012 -> 2014
  PM at Viber

- 2014 -> 2016
  VP of Product at Buynando

- 2016 -> NOW
  MA at NYU

*** Programming in Python for 2 years**

LUNCZ: PLAYING DATA

WHAT'S LUNCZ?: MUSIC AS A TOOL

- A tool that generates synthesized music based live music played on an acoustic or an amplified instrument

- Luncz generates music that accommodates the player and allow further development of a musical idea

LETS SEE IT!

INGREDIENTS

- **PyAudio**
  https://people.csail.mit.edu/hubert/pyaudio/

- **LibROSA**
  librosa.github.io/librosa/index.html

- **Csound**
  http://www.csounds.com/

- **ctcsound**
  https://github.com/fggp/ctcsound

HOW DOES IT WORK?

INPUT

```
source = './_recordings/'
destination = './_recordings/backup/'

# AUDIO STREAM CONFIGURATIONS
CHUNK = 1024
FORMAT = pyaudio.paInt16
CHANNELS = 1
RATE = 44100
RECORD_SECONDS = 10
AUDIO_OUTPUT_TYPE = ".wav"
WAVE_OUTPUT_FILENAME_NO_EXTENSION = "_recordings/" + datetime.datetime.now().isoformat()
WAVE_OUTPUT_FILENAME = "_recordings/" + datetime.datetime.now().isoformat() + AUDIO_OUTPUT_TYPE

audio = pyaudio.PyAudio()

# STARTING THE STREAM
stream = audio.open(format=FORMAT,
                    channels=CHANNELS,
                    rate=RATE,
                    input=True,
                    frames_per_buffer=CHUNK)

frames = []

for i in range(0, int(RATE / CHUNK * RECORD_SECONDS)):
    data = stream.read(CHUNK)
    frames.append(data)

# CLOSING THE STREAM
stream.stop_stream()
stream.close()
audio.terminate()

# SAVING STREAM TO AUDIO FILE
wf = wave.open(WAVE_OUTPUT_FILENAME, 'wb')
wf.setnchannels(CHANNELS)
wf.setsampwidth(audio.get_sample_size(FORMAT))
wf.setframerate(RATE)
wf.writeframes(b''.join(frames))
wf.close()
```

AUDIO ANALYSIS

```python
"""""""""""""""""""""""""""""""""""""""
3 - get notes
"""""""""""""""""""""""""""""""""""""""
hz = librosa.feature.chroma_cqt(y=y, sr=sr)

## GET STRONGEST OCTAVE
strongest_octave = 0
strongest_octave_sum = 0
for octave in range(len(hz)):
    sum = 0
    for frame in hz[octave]:
        sum = sum + frame
    if sum > strongest_octave_sum:
        strongest_octave_sum = sum
        strongest_octave = octave

## GET HEIGHEST HZ FOR EACH TIME FRAME
strongest_hz = []
for i in range(len(hz[0])):
    strongest_hz.append(0)

notes = []
for i in range(len(hz[0])):
    notes.append(0)

for frame_i in range(len(hz[0])):
    strongest_temp = 0
    for octave_i in range(len(hz)):

        if hz[octave_i][frame_i] > strongest_temp:
            strongest_temp = hz[octave_i][frame_i]
            strongest_hz[frame_i] = octave_i + 1
            notes[frame_i] = librosa.hz_to_note(hz[octave_i][frame_i])
```

OUTPUT: CSOUND

```
;-----------------------------------------------------------------------
; Instrument: Background music
;-----------------------------------------------------------------------

instr 102

iduration = p3
iattack = 0.5
icurrentnote = 0
inoteC = cpspch (6.0)
inoteG = cpspch (6.07)
inoteD = cpspch (5.2)
itimbre = p7

kNotesArray[] init 3
kNotesArray[] fillarray inoteC, inoteG, inoteD
alfo = 0

if (gifreq1 == 0) then
     kfreq = kNotesArray[p6]
     iamplitude = 0.1
     itable1 = 2
     itable2 = 2
else
     icurrentnote = gifreq1
     kfreq = cpspch (icurrentnote)
     iamplitude = 0.1
     itable1 = 3
     itable2 = 2
     alfo lfo 2, 8
endif


if (p6 == 0) && (p5 == 1) then
     schedule p1, 12*60/gitempo*0.8, 12*60/gitempo, iamplitude, 1 ,1, itable1
     schedule p1, 12*60/gitempo*0.8, 12*60/gitempo, iamplitude, 0 ,1, itable2

elseif (p6 == 1) && (p5 == 1)  then
     schedule p1, 12*60/gitempo*0.8, 12*60/gitempo, iamplitude, 1 ,2, itable1
     schedule p1, 12*60/gitempo*0.8, 12*60/gitempo, iamplitude, 0 ,2, itable2

elseif (p6 == 2) && (p5 == 1)  then
     schedule p1, 12*60/gitempo*0.8, 12*60/gitempo, iamplitude, 1 ,0, itable1
     schedule p1, 12*60/gitempo*0.8, 12*60/gitempo, iamplitude, 0 ,0, itable2

endif

aenv linseg 0, iduration * iattack, p4, iduration * ( 1 - iattack ), 0


ares poscil3 aenv, kfreq+alfo, itimbre
ares moogvcf ares, 400, 0.8
outs ares, ares

endin
```

OUTPUT: CSOUND

```
<CsScore>

f1 0 [2^16] 10 1 1 0.05 0 ; Sine
f2 0 [2^16] 10 1 0.15 6 2 1 ; default background sound
f3 0 [2^14] 10 1 0   0.3 0    0.2 0     0.14 0     .111
f4 0 [2^14] 10 1 1   1   1    0.7 0.5   0.3  0.1

i 1020    12        0.2       1          0    2
i 103     0     99999999999


</CsScore>
```

OUTPUT: PYTHON

```
for row in csv_file:
    beat_list.append(row[0])
open_file.close()

for beat in beat_list:
    original_beat_times.append(float(beat))

n = 1
print ('* * * * * * * * * * * * * * * * * * * * * * *')
print ('SENDING TO CSOUND')
print ('* * * * * * * * * * * * * * * * * * * * * * *')
print ('Getting data from:', file)
print ('Beat onset times:', original_beat_times)
print ('Csound score lines:')
for time in original_beat_times:

    s_per_beat = 60 / recording_tempo
    s_per_measure = s_per_beat * len(beat_list)
    loop_length = s_per_measure * 1

    modified_time = recording_tempo*time/60

    if (loop_length <= 6) and (time == original_beat_times[len(original_beat_times)-1]):
        continue

    if (loop_length - modified_time) < 0:
        continue

    if 6 - modified_time < 0.8:
        pt.scoreEvent(False, 'i', (100, modified_time, 1, 0, cpspch_array[data[n]], version, 1, recording_tempo, loop_length))
        print (100, modified_time, 1, 0, cpspch_array[data[n]], version, 1, recording_tempo, loop_length)
    else:
        pt.scoreEvent(False, 'i', (100, modified_time, 1, 0.2, cpspch_array[data[n]], version, 1, recording_tempo, loop_length))
        print (100, modified_time, 1, 0.2, cpspch_array[data[n]], version, 1, recording_tempo, loop_length)
```

LIVE DEMO

WANT TO KNOW MORE?

- **Reach out!**

  dror.ayalon@nyu.edu

- **Hop on GitHub**

  https://github.com/dodiku/Luncz

- **See my new portfolio**

  https://www.drorayalon.com/#/luncz/

Thanks 🙏