รายงาน

Interactive Musical Instruments / Synthesis with Python

จัดทำโดย

ณัฐกฤตย์ จตุภัทร์ดิษฐ์	6101012610037
เดชมนต์ แจ้งจิตต์	6101012610061
เธษฐา สุภโตษะ	6101012620091
วศวัตต์ นนท์ธีระวิชยา	6101012620148

เสนอ

อาจารย์ เรวัต ศิริโภคาภิรมย์

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา Introduction to Signal

สัญญาณเบื้องต้น รหัสวิชา 010123106

ภาคเรียนที่ 1 ปีการศึกษา 2563

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชาภาษาไทยเพื่อกาเรียนรู้ในระดับชั้นปริญญาตรีชั้นปีที่ 3 โดยมีจุดประสงค์เพื่อการศึกษาความรู้ที่ได้จากเรื่องระบบและสัญญานเบื้องต้นรวมไปถึงการประยุกต์ใช้กับภาษาไพธอนทั้งนี้ในรางานฉบับนี้มีเนื้อหาซึ่งประกอบด้วยความรู้เกี่ยวกับสัญญานคลื่นความถี่ของเสียงและมาตรฐานการประสานเครื่องดนตรีแบบดิจิตอลหรือมิดิที่นำมาประยุกต์ใช้เป็นซอฟต์แวร์เสียงสังเคราะห์ ตามหัวข้อ Interactive Musical Instruments / Sysnthesis with Python

ผู้จัดทำหวังว่ารายงานเล่มนี้จะเป็นประโยชน์ในในเรื่องการสร้างแอพลิเคชันเสียงสังเคราะห์จากภาษาไพธอน หวังว่ารายงานฉบับนี้จะให้ความรู้ และเป็นประโยชน์แก่ผู้อ่านทุก ๆ ท่าน หากมีข้อเสนอแนะประการใด คณะผู้จัดทำขอรับไว้ด้วยความขอบพระคุณยิ่ง

ขอแสดงความนับถือ

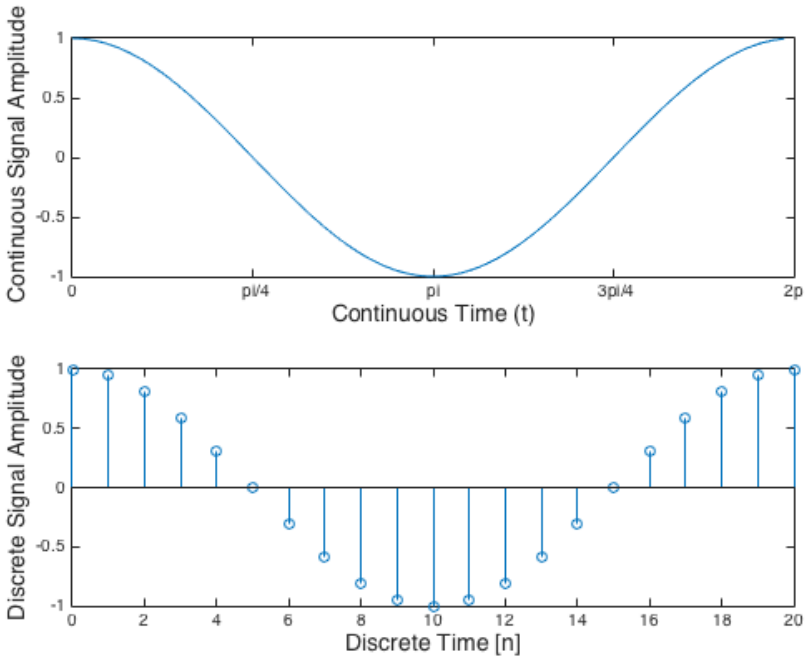
คณะผู้จัดทำ

GitHub Project https://github.com/best2000/synth_key

การสังเคราะห์เสียงด้วย Python + NumPy and Pyo Library

ในการที่จะสร้างเสียงให้เกิดขึ้นมาได้นั้นในทางธรรมชาติเสียงจะเกิดขึ้นได้จากการการที่มีคลื่นเสียงเกิดขึ้น ซึ่งคลื่นเสียงนั้นจะมีลักษณะเป็นคลื่น Sine ซึ่งเป็นคลื่นแบบ continuous signal หรือเป็นสัญญาณที่มีความต่อเนื่องทางเวลา ซึ่งเป็นสัญญาณแบบอนาล็อก

แต่ว่าในอุปกรณ์ดิจิตอลนั้น การที่จะจัดเก็บคลื่นสัญญาณแบบ continuous นั้น จะไม่สามารถจัดเก็บในรูปแบบของสัญญาณต้นฉบับได้ เนื่องจากการทำงานของอุปกรณ์ดิจิตอล จะสามารถเก็บข้อมูลสัญญาณได้ในแบบ Discrete Signal ซึ่งเป็นการเก็บค่าแอมปลิจูดที่ตำแหน่งเวลา โดยมีช่วงระยะห่างที่เรียกว่า Sampling Rate



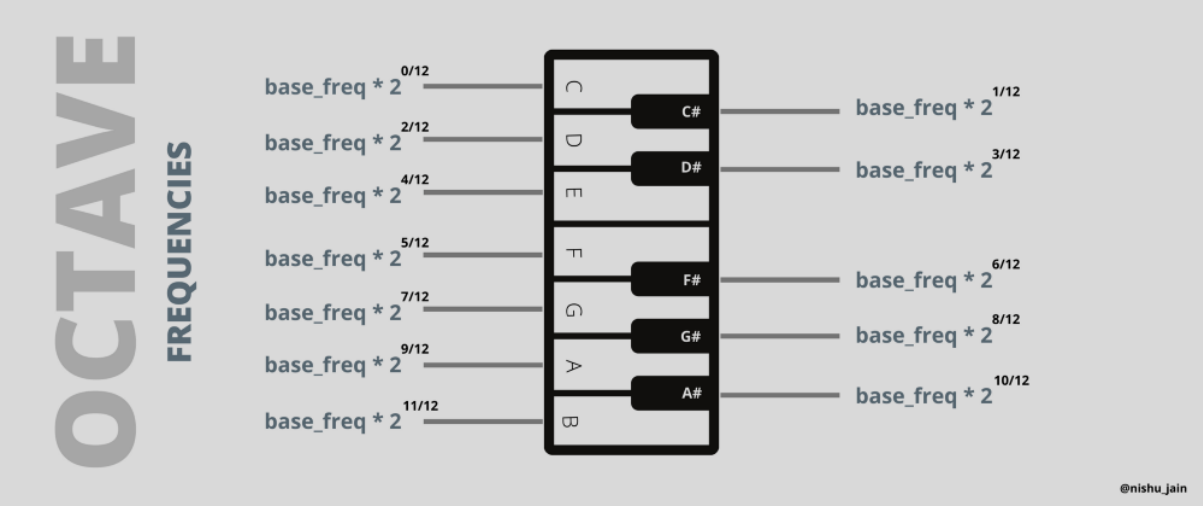
จากภาพ จะเห็นว่าการจัดเก็บสัญญาณในรูปแบบ Discrete Time นั้น จะใช้การเก็บ Amplitude ที่ตำแหน่งเวลาซึ่งการที่อุปกรณ์ดิจิตอลจะสามารถส่งสัญญาณที่เป็นอนาล็อกออกมาได้นั้นจะต้องใช้อุปกรณ์ที่เรียกว่า Digital to Analog Converter (DAC) เพื่อสร้างสัญญาณอนาล็อกขึ้นมาใหม่

ดังนั้น ในการสังเคราะห์สัญญาณเสียงด้วย Python เราจะต้องทำการสร้างชุดข้อมูลแบบ Discrete ขึ้นมา โดยการใช้ NumPy ตามวิธีการคำนวณด้านล่าง และนำ Array ของ Amplitude ที่ทำการคำนวณด้วย NumPy แล้ว มาสังเคราะห์เป็นเสียงด้วย Pyo

หลักการคำนวนความถี่ของแต่ละ key

เราสามารถหาความถี่เริ่มต้นได้จากตารางความถี่ของโน๊ตแต่ละตัว

Scientific designation	Helmholtz designation	Octave name	Frequency (Hz)	Other names	Audio
C_{-1}	C,,, or ,,,C or CCCC	Octocontra	8.176		Play (help·info)
C_0	C,, or ,,C or CCC	Subcontra	16.352		Play (help·info)
C_1	C, or ,C or CC	Contra	32.703		Play (help·info)
C_2	C	Great	65.406	Low C, cello C, 8′ C (see organ pipe length)	Play (help·info)
C_3	c	Small	130.813	4′ C or tenor C (organ), viola C	Play (help·info)
C_4	c′	One-lined	261.626	**Middle C**	Play (help·info)
C_5	c″	Two-lined	523.251	Treble C, high C (written an octave higher for tenor voices)[4]	Play (help·info)
C_6	c‴	Three-lined	1,046.502	High C (soprano)	Play (help·info)
C_7	c⁗	Four-lined	2,093.005	Double high C[citation needed]	Play (help·info)
C_8	c⁗′	Five-lined	4,186.009	Eighth octave C, triple high C	Play (help·info)
C_9	c⁗″	Six-lined	8,372.018	Quadruple high C	Play (help·info)
C_{10}	c⁗‴	Seven-lined	16,744.036	Quintuple high C	Play (help·info)



(base_freq)*[(2)^(note_seq/12)]

โดยโน๊ตตัวต่อไปจะนำ base_freq * 2 ยกกำลังตามตำแหน่งโน๊ตส่วนโน๊ตทั้งหมด (12) เรียงจากตัวแรกเริ่ม 0 ถึงตัวสุดท้ายคือ 12

Samplerate = 44100

```
20  def get_wave(freq, duration=0.5):
21      amplitude = 4096
22      t = np.linspace(0, duration, int(samplerate * duration))
23      wave = amplitude * np.sin(2 * np.pi * freq * t)
24
25      return wave
26
27
28  def get_song_data(music_notes):
29      note_freqs = get_piano_notes()
30      song = [get_wave(note_freqs[note]) for note in music_notes.split('-')]
31      song = np.concatenate(song)
32      return song.astype(np.int16)
```

จากนั้น เมื่อคำนวณได้ความถี่ของแต่ละโน๊ตมาแล้ว จะต้องทำการคำนวณค่า Amplitude โดยจะใช้ Function np.sin ใน NumPy Library

```
20  def get_wave(freq, duration=0.5):
21      amplitude = 4096
22      t = np.linspace(0, duration, int(samplerate * duration))
23      wave = amplitude * np.sin(2 * np.pi * freq * t)
24
25      return wave
26
27
28  def get_song_data(music_notes):
29      note_freqs = get_piano_notes()
30      song = [get_wave(note_freqs[note]) for note in music_notes.split('-')]
31      song = np.concatenate(song)
32      return song.astype(np.int16)
```

Function นี้จะสร้าง Array ของคลื่น Sine ตามความถี่และระยะเวลาที่กำหนด โดยจะกำหนดจำนวน Sample ตาม Sample Rate * duration ดังนั้น ถ้ากำหนด duration = 0.5 จะหมายถึงการสร้าง array ของสัญญาณแบบ Discrete ที่มีระยะเวลา 0.5 วินาที โดยใน Array จะมี 0.5 * 44100 = 22050 samples

จาก t = np.linspace(0, duration, int(samplerate * duration))

เราจะใช้ ฟังก์ชั่น np.linspace ในการสร้างตัวแปร t จากการกำหนดค่าระยะจาก 0 ถึง duration โดยมีจำนวนข้อมูลตามจำนวน sample

หลังจากนั้นจะทำการสร้างคลื่นจากสมการ Wave = Asin($2\Pi ft$)

โดย A คือ amplitude

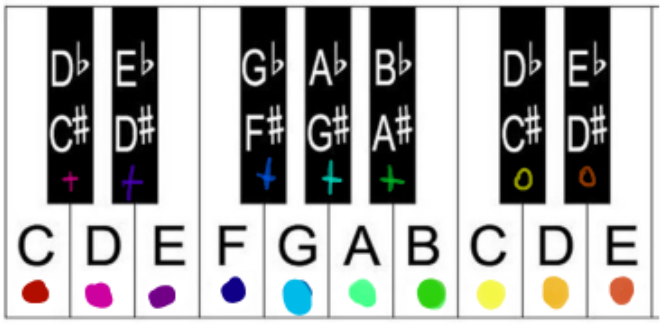
Python Synth Keyboard มี feature ดังนี้

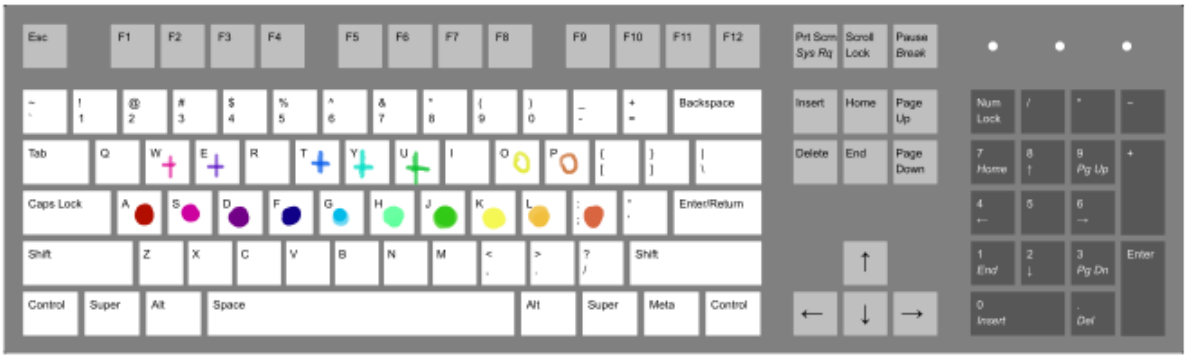
- เล่นเสียง synth จากคอมพิวเตอร์
- ปรับ Octave Key ได้
- เปลี่ยนรูปกราฟ คลื่นเสียง หรือ อินพุตเพื่อได้เอาท์พุตตามต้องการได้

How it is made เราได้ใช้ 2 libraries หลักๆคือ

- pyo มีหน้าที่ generate เสียงสังเคราะห์
- pynput มีหน้าที่ รับ input จาก keyboard

แผนผัง key mapping ของโปรแกรม





ปุ่มโหมดต่างๆ

ปุ่ม Z : Sine wave

ปุ่ม X : Supersaw

ปุ่ม C : Supersaw mixing

ปุ่ม V : Triangle

ปุ่ม B : Square

ปุ่มในการกำหนด Octave

ปุ่ม “0” = ห้องเสียงที่มีความถี่เริ่มต้น 16.35 Hz

ปุ่ม “1” = ห้องเสียงที่มีความถี่เริ่มต้น 32.7 Hz

ปุ่ม “2” = ห้องเสียงที่มีความถี่เริ่มต้น 65.41 Hz

ปุ่ม “3” = ห้องเสียงที่มีความถี่เริ่มต้น 130.81 Hz

ปุ่ม “4” = ห้องเสียงที่มีความถี่เริ่มต้น 261.63 Hz

ปุ่ม “5” = ห้องเสียงที่มีความถี่เริ่มต้น 523.25 Hz

ปุ่ม “6” = ห้องเสียงที่มีความถี่เริ่มต้น 1046.50 Hz

ปุ่ม “7” = ห้องเสียงที่มีความถี่เริ่มต้น 2093 Hz

ปุ่ม “8” = ห้องเสียงที่มีความถี่เริ่มต้น 4186.01 Hz

ปุ่ม “9” = ห้องเสียงที่มีความถี่เริ่มต้น 8372.019 Hz

โดยมีหลักการดังนี้

1. Generate คลื่นเสียงขึ้นมา
2. รับ Keyboard Input
3. Input แต่ละตัวจะมีค่าความถี่ที่แตกต่างกัน
4. Output ออกมาเป็นเสียงโน๊ตดนตรี

รายการอุปกรณ์ (ซอฟต์แวร์และฮาร์ดแวร์) ที่จะนำมาใช้งาน

Software

ภาษา Python Version 3.8 โดยมี Libary ดังนี้

- pyo version 1.0.3
- pynput

Hardware

- คอมพิวเตอร์และโน๊ตบุค
- computer keyboard

ปัญหาที่พบ

ไม่สามารถรับอินพุตพร้อมกันมากกว่าหนึ่งปุ่มพร้อมกันไม่ได้ซึ่งเราทดลองการเขียนโปรแกรมแบบ multiprocessing ซึ่งพบเออเรอร์จำนวนมากและการใช้งาน thread ให้ประสานกันตรงพอดีนั้นยังไม่สามารถทำได้

อ้างอิง

pyo tutorial

Chrome music lab

Music Synthesis in Python

Note Frequency