

コンピュータで何をすればよいか？音を出してみよう
文責：高知大学名誉教授 中村 治

Pygame.midi を使って電子ピアノを作ってみよう！

Python のバージョンに適合した Pygame をインストールします。

```
import pygame.midi
import time
pygame.midi.init()
player = pygame.midi.Output(0)
player.set_instrument(0)
player.note_on(64, 127)
time.sleep(1)
player.note_off(64, 127)
pygame.midi.quit()
```

で、ひとまず音がでます。更に、

```
import pygame
import pygame.midi
import time
pygame.midi.init()
player = pygame.midi.Output(0)
player.set_instrument(48,1)
major = [0,4,7,12]
def go(note):
    player.note_on(note, 127,1)
    time.sleep(1)
    player.note_off(note,127,1)
def arp(base, ints):
    for n in ints:
        go(base+n)
def chord(base, ints):
    player.note_on(base,127,1);
    player.note_on(base+ints[1],127,1)
    player.note_on(base+ints[2],127,1)
    player.note_on(base+ints[3],127,1)
    time.sleep(1)
    player.note_off(base,127,1);
    player.note_off(base+ints[1],127,1)
    player.note_off(base+ints[2],127,1)
```

```

        player.note_off(base+ints[3],127,1)
def end():
    pygame.quit()
go(60)
chord(60, major)
arp(60, major)
end()

```

と言うサンプルプログラムもインターネットで見つかります。

これらのプログラムの意味を理解するために

<https://www.pygame.org/docs/ref/midi.html> の

Pygame.midi---Pygame v1.9.2 documentation を覗いてみます。必要なのは

```

player.set_instrument(0)
player.note_on(64, 127)
player.note_off(64, 127)

```

の意味です。

```

set_instrument()
    select an instrument, with a value between 0 and 127
    set_instrument(instrument_id, channel=0) -> None

```

と書いてあるので、これで鳴らしたい楽器を指定しています。0 から 1 2 7 まで 1 2 8 種類の楽器を指定出来るみたいです。

```

note_on()
    turns a midi note on. Note must be off.
    note_on(note, velocity=None, channel=0) -> None
    Turn a note on in the output stream.
    The note must already be off for this to work correctly.

```

と書いてあるので、note(音符) と velocity(速度が日本語訳ですが音量と違っていいと思います) を指定しています。

```

note_off()
    turns a midi note off. Note must be on.
    note_off(note, velocity=None, channel=0) -> None
    Turn a note off in the output stream.
    The note must already be on for this to work correctly.

```

と書いてあるので、note_on() と note_off() をペアーで使わなければなりません。

使い方が分かったので、電子ピアノを作ってみましょう。誰かすでにいいものを作ってくれていれば自分で作る必要はありません。インターネットで探してみます。

<http://d.hatena.ne.jp/kadotanimitsuru/20100112/midi> の

Pygame で MIDI - つちのこ、のこのこ。(はてな番外地) に Python のプログラムがあります。

```

# -*- coding:utf-8 -*-

''' MIDI keyboard
マウス左ボタンで発音、右ボタンで完全 5 度を同時に発音。
スペースバーを押しながらだとサスティン。
鍵盤の下の方を押すほど大音量
'''

import pygame
import pygame.midi
from pygame.locals import *
INSTRUMENT = 0 # 楽器の種類 (0-127)
# 0: Piano, 19:Organ, 56:Trumpet, 91:Voice 等
KEY_WIDTH = 20 # 鍵盤の幅
WIDTH, HEIGHT = 800, 128 # 画面の大きさ
FPS = 60
NOTE_CENTER = 60 # 中央の音。C(ド) の音を指定
COLOR = 0, 255, 200 # 色
WHITE_COLOR = 255, 255, 255 # 白鍵の色
BLACK_COLOR = 0, 0, 50 # 黒鍵の色
BG_COLOR = 100, 0, 50 # 背景色
KEY_COLOR = 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0 # 0 = 白鍵 1 = 黒鍵
NOTE_NAME = ('C', 'C#', 'D', 'D#', 'E', 'F', 'F#', 'G', 'G#', 'A', 'A#', 'B')

def main():
    pygame.init()
    pygame.midi.init()
    screen = pygame.display.set_mode((WIDTH, HEIGHT))
    midiout = pygame.midi.Output(pygame.midi.get_default_output_id())
    midiout.set_instrument(INSTRUMENT)
    clock = pygame.time.Clock()
    clock.tick(FPS)
    keys = WIDTH // KEY_WIDTH + 1
    keylist = [False] * (keys + 7)
    note_start = NOTE_CENTER - keys // 2
    note_no = None
    vel = 0
    sustain = False
    while True:
        for e in pygame.event.get():
            if e.type is QUIT:
                return
            elif e.type is KEYDOWN and e.key is K_ESCAPE:

```

```

        return
    elif e.type is KEYDOWN and e.key is K_SPACE:
        sustain = True
    elif e.type is KEYUP and e.key is K_SPACE:
        sustain = False
        note_no = None
        for key, b in enumerate(keylist):
            if b:
                midiout.note_off(note_start + key, 0)
                keylist[key] = False
    elif e.type is MOUSEBUTTONDOWN and (e.button == 1 or e.button == 3):
        x, y = e.pos
        vel = 128 * y // HEIGHT
        key = x // KEY_WIDTH
        keylist[key] = True
        note_no = note_start + key
        midiout.note_on(note_no, vel)
        if e.button == 3:
            keylist[key + 7] = True
            midiout.note_on(note_no + 7, vel)
    elif e.type is MOUSEBUTTONUP and (e.button == 1 or e.button == 3):
        if not sustain:
            note_no = None
            for key, b in enumerate(keylist):
                if b:
                    midiout.note_off(note_start + key, 0)
                    keylist[key] = False
    screen.fill(BG_COLOR)
    for key in range(keys):
        x = key * KEY_WIDTH
        pygame.draw.rect(screen,
            (WHITE_COLOR, BLACK_COLOR)[KEY_COLOR[(note_start + key) % 12]],
            (x + 1, 0, KEY_WIDTH - 2, HEIGHT))
        if keylist[key]:
            pygame.draw.rect(screen, COLOR, (x, 0, KEY_WIDTH, HEIGHT), 3)
    clock.tick(FPS)
    pygame.display.flip()
    notes = []
    for key, b in enumerate(keylist):
        if b:
            nn = note_start + key
            notes.append(' [0]:[1]'.format(NOTE_NAME[nn % 12], nn))
    pygame.display.set_caption(', '.join(notes))

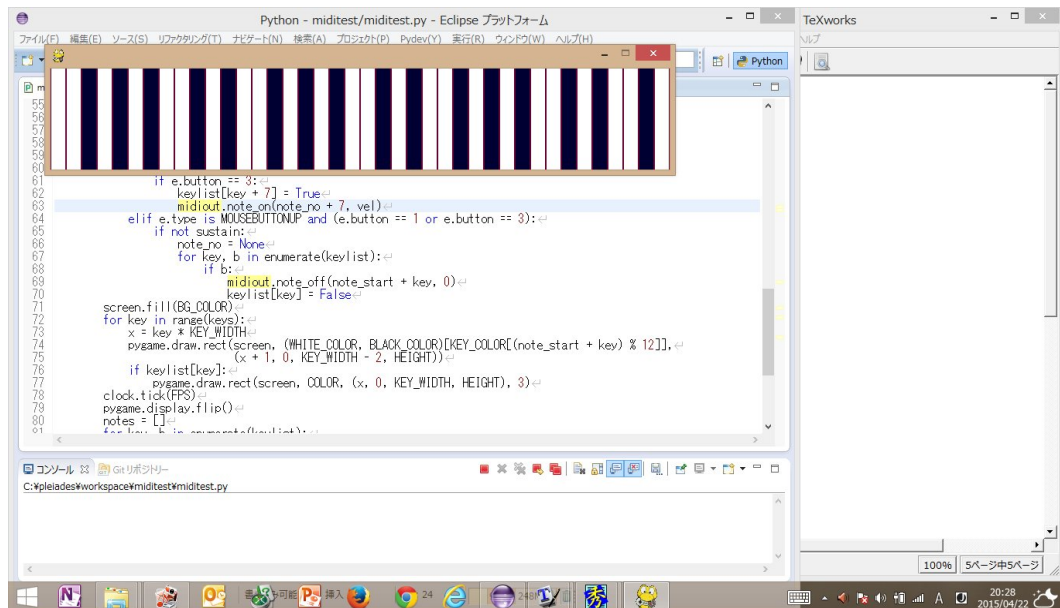
```

```

if __name__ == '__main__':
    try:
        main()
    finally:
        pygame.quit()

```

実行すると



です。ちゃんと電子ピアノになっています。素晴らしい！これがあれば別に自分で作ることはないですが、楽器を変えるにはプログラムをいじらなければいけないことと鍵盤が鍵盤らしくない事です。



のような鍵盤にしたいです。上のプログラムを真似して、Pygame の画面ではなく、Tkinter の画面でそれらしきものを作ってみましょう。

```

# -*- coding:utf-8 -*-

''' MIDI keyboard
鍵盤の下の方を押すほど大音量
'''

from Tkinter import *
import pygame.midi
from pygame.locals import *
import time

INSTRUMENT = 0 # 楽器の種類 (0-127)
# 0: Piano, 19:Organ, 56:Trumpet, 91:Voice 等
WIDTH, HEIGHT = 920, 128 # 画面の大きさ
KEY_WIDTH = WIDTH // 23 # 鍵盤の幅
NOTE_CENTER = 60 # 中央の音。C(ド) の音を指定
BLACK_KEY = (1, 0, 1, 1, 0, 1, 1) * 4 # 0 = 白鍵 1 = 黒鍵

pygame.midi.init()
midiout = pygame.midi.Output(pygame.midi.get_default_output_id())
midiout.set_instrument(INSTRUMENT)
keys = WIDTH // KEY_WIDTH
note_start = NOTE_CENTER - 14
note_no = None
vel = 0

def ButtonPress(event):
    mX = event.x
    mY = event.y
    index = -1
    val = 0
    for key, b in enumerate(BLACK_KEY):
        if b:
            val += 2
        else:
            val += 1
        if b:
            if key == 0:
                if mX < KEY_WIDTH//3 and mY < 3*HEIGHT//5:
                    note_no = note_start
                    vel = 128 * mY * 5 // (3 * HEIGHT)
                    midiout.note_on(note_no, vel)
                    index = key
                    print note_no

```

```

        else:
            x = key*KEY_WIDTH-KEY_WIDTH//3
            if x<mX and mX<x+2*KEY_WIDTH//3 and mY < 3*HEIGHT//5:
                note_no = note_start + val - 2
                vel = 128 * mY * 5 // (3 * HEIGHT)
                midiout.note_on(note_no, vel)
                index = key
                print note_no
    if index == -1:
        val = 0
        for key in range(keys):
            if BLACK_KEY[key]:
                val += 2
            else:
                val += 1
        x = key*KEY_WIDTH
        if x<mX and mX<x+KEY_WIDTH:
            note_no = note_start + val - 1
            vel = 128 * mY // HEIGHT
            midiout.note_on(note_no, vel)
            index = key
            print note_no
    if index == -1:
        return

def ButtonRelease(event):
    mX = event.x
    mY = event.y
    index = -1
    val = 0
    for key, b in enumerate(BLACK_KEY):
        if b:
            val += 2
        else:
            val += 1
    if b:
        if key == 0:
            if mX < KEY_WIDTH//3 and mY < 3*HEIGHT//5:
                note_no = note_start
                vel = 128 * mY * 5 // (3 * HEIGHT)
                midiout.note_off(note_no, vel)
                index = key
            else:

```

```

        x = key*KEY_WIDTH-KEY_WIDTH//3
        if x<mX and mX<x+2*KEY_WIDTH//3 and mY < 3*HEIGHT//5:
            note_no = note_start + val - 2
            vel = 128 * mY * 5 // (3 * HEIGHT)
            midiout.note_off(note_no, vel)
            index = key
    if index == -1:
        val = 0
        for key in range(keys):
            if BLACK_KEY[key]:
                val += 2
            else:
                val += 1
        x = key*KEY_WIDTH
        if x<mX and mX<x+KEY_WIDTH:
            note_no = note_start + val - 1
            vel = 128 * mY // HEIGHT
            midiout.note_off(note_no, vel)
            index = key
    if index == -1:
        return

def SetInstrument():
    a = int(content.get())
    midiout.set_instrument(a)

def EndMidi():
    pygame.midi.quit()

root = Tk()
canvas = Canvas(root, width=WIDTH, height=HEIGHT)
canvas.pack()
canvas.bind("<ButtonPress-1>", ButtonPress)
canvas.bind("<ButtonRelease-1>", ButtonRelease)
button = Button(root, text='INSTRUMENT SET', command=SetInstrument).pack(side = LEFT)
content = StringVar()
entry = Entry(root, textvariable=content).pack(side=LEFT)
canvas.create_rectangle(0, 0, WIDTH, HEIGHT, fill='blue')
for key in range(keys):
    x = key * KEY_WIDTH
    canvas.create_rectangle(x + 1, 0, x+KEY_WIDTH - 1, HEIGHT, fill='white')
for key, b in enumerate(BLACK_KEY):
    if b:

```



```

if key == 0:
    x = key * KEY_WIDTH
    canvas.create_rectangle(x, 0, x+KEY_WIDTH//3 - 1, 3*HEIGHT//5, fill='black')
else:
    x = key * KEY_WIDTH - KEY_WIDTH//3
    canvas.create_rectangle(x + 1, 0, x+2*KEY_WIDTH//3 - 1, 3*HEIGHT//5, fill='black')

content.set(0)
button = Button(root, text='END MIDI', command=EndMidi).pack(side = LEFT)

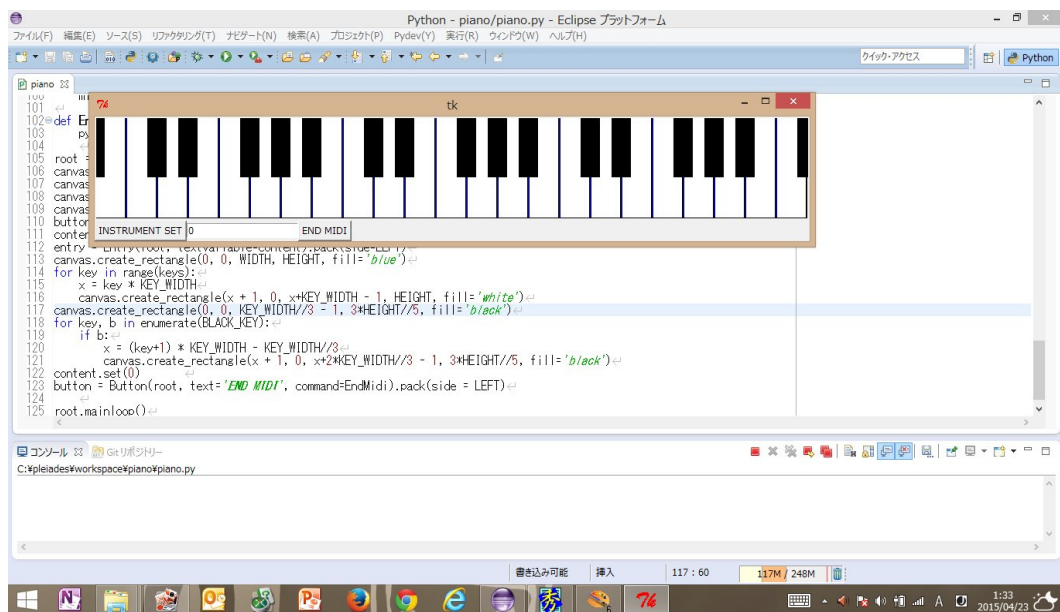
root.mainloop()

```

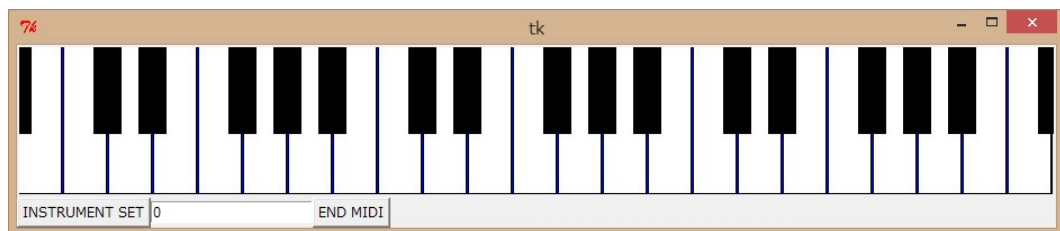
がプログラムです。

右ボタンで完全5度を同時に発音。
スペースバーを押しながらだとサステイン。

は、音楽は苦手で、何のこともまるっきり分からなく、必要性が理解できないので省略しました。
実行すると



です。



のエントリー（テキストボックスのこと）に 0 から 127 の数字を入れて、「INSTRUMENT SET」のボタンをクリックすると楽器を変えることができます。「END MIDI」のボタンで Midi を終了します。

Exception Exception: "PortMidi: 'Bad pointer'" in <pypm.Output object at 0x02926C80> ignored

のメッセージが終了後に出ますが、私にはこの意味が分からないので無視します。調べるとマウスやキーを使うとこのようなメッセージが出るみたいですが、如何すればこのメッセージが出なくなるか分かりませんが、多分そのままでも無害です。また、意欲のある人は、キーボードを鍵盤代わりにするプログラムをプログラミングしてください。Midi ファイルを読み込んで、演奏させたり、Midi ファイルを何らかの方法で作って、保存したりも出来るはずです。興味があれば、自分で調べて、作ってください。インターネットで必要な情報が得られるはずです。