

合成音乐

- import json
- import numpy as np
- from scipy.io.wavfile import write
- import matplotlib.pyplot as plt
- ''' 功能作用：合成音乐
- 文件：tone_freq_map.json的JSON文件，该文件包括一些音阶以及它们的频率：
- 步骤：
 - 1.开始读取json的频率映射文件
 - 2.设置生成G调的输入参数
 - 3.开始生成音阶
 - 先定义生成音调的函数
 - 1.创建时间轴
 - 2.构建音频信号,参数有: 幅度, 频率
 - 3.返回一个数据类型是16位的音阶数据
 - 4.将生成的信号写入输出文件中,文件中需要写入的是wav格式,传递的参数有: HZ频率 和 音阶
 - 5.构建音阶以及持续的时间,代表着,每个音可以持续听得到的声音多长 为tone_seq
 - 6.然后通过tone_seq 提取出每个音阶和时间,传递给synthesizer,生成音调函数处理
 - 7.再输出的文件,写入输出文件中,结束.
- '''# 定义一个函数, 该函数基于输入参数合成音调# 定义合成音调
- def synthesizer(freq, duration, amp=1.0, sampling_freq=44100):
 - '''
 - :param freq: 频率
 - :param duration: 音频时间长度
 - :param amp: 振幅
 - :param sampling_freq: 采样率
 - :return: 返回指定音调类型的数据
 - ''' # 创建时间轴 X
 - t = np.linspace(0, int(duration), int(duration) * sampling_freq)
 - # 构建音频信号,参数: 幅度和频率
 - audio = amp * np.sin(2 * np.pi * freq * t)
 - return audio.astype(np.int16)
- if __name__ == '__main__':
- # 读取频率映射文件
- tone_map_file = 'tone_freq_map.json'
- with open(tone_map_file, 'r') as f:
- tone_freq_map = json.loads(f.read())
- # 设置生成G调的输入参数

- `input_tone = 'G'`
- `duration = 2` # seconds
- `amplitude = 10000`
- `sampling_freq = 44100` # Hz
- # 生成音阶
- `synthesized_tone = synthesizer(tone_freq_map[input_tone], duration, amplitude, sampling_freq)`
- # 将生成的信号写入输出文件
- `write('output_tone.wav', sampling_freq, synthesized_tone)`
- # 音阶以及持续的时间
- `tone_seq = [('D', 0.3), ('G', 0.6), ('C', 0.5), ('A', 0.3), ('Asharp', 0.7)]`
- # 构建基于和弦序列的音频信号
- `output = np.array([])`
- `for item in tone_seq:`
- `input_tone = item[0]`
- `duration = item[1]`
- `synthesized_tone = synthesizer(tone_freq_map[input_tone], duration, amplitude, sampling_freq)`
- `output = np.append(output, synthesized_tone, axis=0)`
- # 写入输出文件
- `write('output_tone_seq.wav', sampling_freq, output)`