

这里给大家说明一下配置文件 `config.py` 里的每个参数对应的含义，以便大家修改的时候更加明确。

尽量涵盖目前所有的参数，不过会侧重比较重要的参数。

`screen_size` 屏幕尺寸（宽度，高度）

`background_image` 背景图片文件路径

`background_size` 背景图片尺寸

`piano_image` 钢琴图片文件路径

`piano_size` 钢琴图片尺寸

`message_color` 和弦显示的字体颜色，格式为(R, G, B, A)

`fonts_size` 字体大小

`label1_place` 显示当前演奏的音符名称的文字位置

`label2_place` 显示和弦名称的文字位置

`label3_place` 播放 midi 文件的状态文字位置

`label_anchor_x` 文字的横向对齐方式

`label_anchor_y` 文字的纵向对齐方式

`fonts =` 字体名称

`bold =` 是否加粗

`notes_image` 音符点模式下音符点的图片路径

`notes_resize_num` 音符点的缩放大小

`go_back_image` 返回按钮的图片文件路径

`go_back_place` 返回按钮的图片位置

`self_play_image` 电脑键盘演奏按钮的图片文件路径

`self_play_place` 电脑键盘演奏按钮的图片位置

`self_midi_image` midi 键盘演奏按钮的图片文件路径

`self_midi_place` midi 键盘演奏按钮的图片位置

`play_midi_image` 播放 midi 文件按钮的图片文件路径

`play_midi_place` 播放 midi 文件按钮的图片位置

`key_settings` 电脑键盘对应 88 键的字典，请注意音源路径里的文件必须要全部包含你所设置到的键位

`reverse_key_settings` 88 键对应电脑键盘的字典

`mode` 这个是我在写这个软件的 UI 之前用来设置是电脑键盘演奏，midi 键盘演奏还是播放 midi 文件的，现在这个参数用不到了 2333

`self_device` 也是之前还没写 UI 时用来设置电脑键盘还是 midi 键盘的，现在用不到了

`midi_device_id` 这个参数是接入 midi 设备后（比如 midi 键盘），midi 设备所对应的 id，一般来说只接入 midi 键盘，没开编曲软件

占用的是 1，有开编曲软件和 loopMIDI 的时候是 2，没接入 midi 键盘只开了编曲软件和 loopMIDI 的时候是 1

以下几个都是播放 midi 文件是的电脑键盘快捷键设置

`pause_key` 暂停的按键

`repeat_key` 重复播放的按键

`unpause_key` （暂停中）继续的按键

`exit_key` 退出程序的按键

`pause_key_clear_notes` 当暂停的时候是否清除掉当前演奏的所有音符的显示

`show_key` 在电脑键盘演奏时是否显示电脑键盘的按键名称

`musicsheet` 这个也是之前还没写 UI 时，播放 midi 文件的模式下，这个参数可以放 musicpy 语言的代码，程序会自动播放，现在

这个参数也用不到了

`path` 也是之前用来设置 `midi` 文件的路径的参数，现在有 `UI` 可以选择文件了，这个参数就没用了

这两个是之前用来设置 `midi` 文件要播放的轨道位置

`track_ind`

`track`

`bpm` 以前用来设置曲速（BPM）的参数，现在我写了 `UI` 就没用了

`play_interval` 以前用来设置播放曲子的某一部分，现在我写了 `UI` 就没用了

#接下来几个是 `pygame` 的音频初始化参数，基本上不需要动，`maxinum_channels` 越大，同时演奏多个音符时越不会出现问题

`frequency = 44100`

`size = -16`

`channel = 1`

`buffer = 1024`

`maxinum_channels = 100`

# 总音量大小，最大为 1，最小为 0

`global_volume = 0.6`

`delay` 是否在弹一个音放开之后给音一定的延迟

`delay_time` 延迟的时间（秒）

`touch_interval` 当连续演奏同一个音的时候，音的结束和重新播放的时间间隔，单位为秒

`delay_only_read_current` 当音在延迟播放的时候（音没有被按着），

和弦判断不包括这些还在延迟播放的音，只包括当前正在按着的音

`sound_format` 音源的文件格式（文件后缀）

`sound_path` 音源的文件路径

`show_delay_time` 播放 midi 文件时音符的延迟时间

这几个是和弦判断的乐理逻辑算法的参数，默认的设置适用程度最广泛，如果我要说明这些都是什么意思的话，

那可能要先理解我的算法，因此这一部分等以后我介绍这个算法的时候再说明

`detect_mode = 'chord'`

`inv_num = False`

`rootpitch = 5`

`ignore_sort_from = False`

`change_from_first = True`

`original_first = True`

`ignore_add_from = True`

`same_note_special = False`

`whole_detect = True`

`return_fromchord = False`

`two_show_interval = True`

`poly_chord_first = False` (这个参数设置为 `True` 的时候，  
会在和弦非常复杂的情况下提前当作复合和弦来进行乐理判断，和弦判断的速度会快很多)

`show_change_pitch` 播放 midi 文件时对整首曲子进行升调或者降调（正数 `n` 为升 `n` 个半音，  
负数 `n` 为降 `n` 个半音）

`show_modulation = [原来的音阶, 转调的音阶]` 播放 midi 文件时，对整首曲子进行转调

`config_enable` 在电脑键盘演奏时是否开启功能键

`config_key` 功能键的键位设置，功能键和其他按键搭配可以做到各种不同的功能

`volume_up` 和功能键搭配的按键，让总音量升高

`volume_down` 和功能键搭配的按键，让总音量降低

`volume_change_unit` 总音量每次变化的音量

`change_delay` 和功能键搭配的按键，改变是否延迟

`change_read_current` 和功能键搭配的按键，改变是否只判断当前按着的和弦

`change_pause_key_clear_notes` 和功能键搭配的按键，改变是否在暂停时清除当前演奏的音的显示

`note_place` 音符点模式下，钢琴从左到右全部按键对应的位置

`load_sound` 演奏时是否加载音源以及播放（在和宿主一起使用的时候要设置为 `False`）

`show_chord` 演奏时是否实时通过乐理逻辑分析判断和弦

这些是音程的名称以及对应的半音数

`perfect_unison = 0`

`minor_second = 1`

`augmented_unison = 1`

`major_second = 2`

`diminished_third = 2`

`minor_third = 3`

`augmented_second = 3`

`major_third = 4`

`diminished_fourth = 4`

`perfect_fourth = 5`

`augmented_third = 5`

`diminished_fifth = 6`

`augmented_fourth = 6`

`perfect_fifth = 7`

`diminished_sixth = 7`

`minor_sixth = 8`

`augmented_fifth = 8`

`major_sixth = 9`

`diminished_seventh = 9`

`minor_seventh = 10`

`augmented_sixth = 10`

`major_seventh = 11`

`diminished_octave = 11`

`perfect_octave = 12`

`octave = 12`

`augmented_seventh = 12`

这些是我写的一首曲子分离主旋律的算法的参数

`melody_tol = minor_seventh`

`chord_tol = major_sixth`

这些是音符条模式下的一些参数

`note_mode` 选择音符显示模式，目前有音符点和音符条（上升）和音符条（下落，只有播放 midi 文件模式可用）三种模式可以选择，

分别对应的是'`dots`'和'`bars`'和'`bars drop`'

`bar_width` 音符条的宽度

`bar_height` 音符条的长度

`bar_color` 音符条的颜色

**bar\_y** 音符条出现的纵坐标

**bar\_offset\_x** 音符条偏离音符点位置的横坐标的像素值

**bar\_opacity** 音符条的透明度，0 到 255，从完全透明到完全不透明

**opacity\_change\_by\_velocity** 透明度是否随着按键力度的变化而变化，

按键力度越轻，音符条越透明，按键力度越重，音符条越不透明

**color\_mode** 音符条的颜色显示模式，目前有单色显示和随机颜色显示两种模式可以选择，

分别对应'normal'和'rainbow'（实际上填不是 normal 的其他文字也可以）

**bar\_steps** 音符条每次上升移动的像素值

**bar\_unit** 在播放 midi 文件的时候，音符条计算相对长度时为单位的长度

**bar\_hold\_increase** 在按住琴键时（或者按住电脑按键时），音符条每次拉长的像素值

**bars\_drop\_interval** 在音符条（下落）模式中，音符条要花多长的时间从屏幕顶端下落到指定位置，单位为秒

**bars\_drop\_place** 在音符条（下落）模式中，音符条下落到的指定位置（高度）

**adjust\_ratio** 调整音符条下落到指定位置的准确度的一个参数，一般来说不需要修改

其他的参数说明

**get\_off\_drums** 如果为 True，在播放 midi 文件模式中，如果选择了合并所有音轨，在读取 midi 文件之后会去掉鼓的轨道，（如果有的话）以避免演示和弦会被鼓的音符扰乱。

**sort\_invisible** 如果为 True，在显示和弦中不会显示排序的内容（比如 "Fmaj7 sort as [2,3,1,4]" 会变成 "Fmaj7"）

**play\_as\_midi** 播放 midi 文件时不加载音源，直接在软件内部播放 midi 文件（以 midi 自带的音源播放），好处是音符比较多的 midi 文件加载快很多，而且播放时在同时播放的音符很多，和弦类型很复杂的时候不会出现卡顿。设置为 True 即可进入这个模式。

**draw\_piano\_keys** 设置为 True 进入绘制钢琴模式，（按照参数和钢琴 88 键的结构绘制出钢琴键盘，取代之前的钢琴图片）在绘制钢琴模式下，midi 键盘演奏或者电脑键盘演奏时对应的琴键会亮起，包括在下落音符模式播放 midi 文件时，音符落在琴键上也会亮起。绘制钢琴是采用直接按照钢琴 88 键的结构，根据可设定的参数绘制出黑白键，并且每个键可以改变

颜色。在 88 键的绘制下面有一张黑色的背景图片，主要用于显示钢琴键之间的缝隙（填充用）。可以关掉音符模式，（`note_mode` 设置为不是 `dots`, `bars`, `bars drop` 的值即可）只打开绘制钢琴模式，演奏时对应的钢琴键位会亮起，播放 `midi` 文件时也会亮起当下的音符。也可以使用任何一个音符模式同时打开绘制钢琴模式。

`white_key_width` 钢琴白键的宽度（横向长度）  
`white_key_height` 钢琴白键的高度（纵向长度）  
`white_key_interval` 每两个钢琴白键之间的距离  
`white_key_y` 钢琴白键的高度位置  
`white_keys_number` 钢琴白键的数量  
`white_key_start_x` 钢琴第一个白键的横向位置  
`white_key_color` 钢琴白键的颜色

`black_key_width` 钢琴黑键的宽度（横向长度）  
`black_key_height` 钢琴黑键的高度（纵向长度）  
`black_key_y` 钢琴黑键的高度位置  
`black_key_first_x` 钢琴第一个黑键的横向位置  
`black_key_start_x` 钢琴第二个黑键的横向位置  
`black_key_color` 钢琴黑键的颜色

`black_keys_set` 除了第 1 个黑键单独设置，每 5 个黑键为一组，每个组里每个黑键之间的相对间隔（第 1 个间隔一般为 0，表示第 1 个黑键从当前组里的相对位置的最左边开始）  
`black_keys_set_interval` 每两个黑键组之间的间隔  
`black_keys_set_num` 黑键组的个数

`piano_background_image` 钢琴底下的背景图片（填充间隙用）

`Ideal Piano` 可以读取一个具有特定格式的作曲分析的文本文件，在演示 `midi` 文件的模式下按照当前小节数来显示曲子的转调，附属和弦，借用和弦等等作曲手法，可以写一个作曲分析的 `txt` 文件，然后实时在当下指定的小节显示对应的作曲分析的内容。默认为 `musical analysis.txt` 这个文件。

作曲分析文件的格式为：

小节数 1 作曲分析内容 1

小节数 2 作曲分析内容 2

小节数 3 作曲分析内容 3

...

这里的小节数以第 1 小节作为开头，小节数为一个数字，可以是整数或者小数。作曲分析内容是你想要在进入到指定的小节数的时候想要显示的内容。软件会解析作曲分析文件的格式，然后寻找第一个到当前小节数的音符的位置，然后在演示曲子的时候当进行到对应的音符的位置的时候显示对应的作曲内容。

目前除了这个格式，还支持显示调性，可以在任意的位置显示当前的调性，如果曲子有转调的时候就可以在转调开始的小节之前写调性的语句。语法为（这里是举例）：

`key:` 调性（调性这里可以写任意的你想显示的内容，比如 A 大调, `A major` 等等）

小节数 1 作曲分析内容 1



小节数 2 作曲分析内容 2

小节数 3 作曲分析内容 3

...

key: 调性（可以在曲子转调的时候写新的调性）

小节数 n 作曲分析内容 n

小节数 x 作曲分析内容 x

小节数 y 作曲分析内容 y

...

（不同小节数与调性语句之间可以空任意多的行，不过在某个小节数内的作曲分析内容必须写成一行，才可以成功解析）

（小节数支持绝对小节位置和相对小节位置两种语法，绝对小节位置就是一个数字，可以是整数或者小数，相对小节位置的语法为+相对小节长度，比如+1 表示相对上一个位置的下一个小节的位置，+1/2 表示相对上一个位置的下一个二分之一小节的位置，相对小节位置支持整数，小数和分数。）

相关参数：

show\_music\_analysis 是否开启显示作曲分析内容

music\_analysis\_file 读取的作曲分析文件的文件路径

music\_analysis\_place 设置显示作曲内容的位置

key\_header 调性开头内容（这个参数是显示调性的开头的内容，比如“当前调性：”）

music\_analysis\_width 音乐分析文本标签的宽度

music\_analysis\_fonts\_size 音乐分析文本的字体大小