

代码结构

仅需要修改 `song` 以及 `long` 所代表的乐曲音高以及音长即可完成乐曲的演奏。本代码使用 `win32api` 进行调用，对 [Freepiano](#) 中的钢琴软件进行曲目的演奏。实现的逻辑比 `pyautogui` 更加高效，支持多线程的按键功能。代码根据使用场景，可选择直接的 `api` 调用以及虚拟按键的调用以适配任务需求（避免部分软件的驱动防作弊功能禁用直接调用 `win32api`）。

直接调用：

需要先实例化一个 `keys` 对象，后进行方法的调用：

```
# keyboard (direct keys)
keys.directKey("a",)
sleep(0.04)
keys.directKey("a", keys.key_release)
```

该代码实现了按下 `a` 按键的功能。

虚拟按键调用：

```
# keyboard (virtual keys)
keys.directKey("a", type=keys.virtual_keys)
sleep(0.04)
keys.directKey("a", keys.key_release, keys.virtual_keys)
```

该代码实现了按下 `a` 虚拟按键的功能。

演奏示例

演奏乐曲小星星，将每一个演奏的音符划分为音高和音长两部分，音高以 `1,2,3,4,6,6,7` 表示一个八度内的7个乐音；以 `1,2,3,4` 等数字表示音长。

通过使用如下函数：

```
keys.directKey(keymap[each[0]])
sleep(each[1])
keys.directKey(keymap[each[0]], keys.key_release)
```

即可模拟人类一根手指所完成的演奏效果。

小星星的乐曲信息如下：

```
# 小星星简谱
song =
[1,1,5,5,6,6,5,4,4,3,3,2,2,1,5,5,4,4,3,3,2,5,5,4,4,3,3,2,1,1,5,5,6,6,5,4,4,3,3,2
,2,1]
# 小星星旋律每个音的音长
long =
[1,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,1,1
,1,2]
```

演奏视频见 `demo.mp4`。个人用户可通过修改song和long的值完成歌曲的自定义演奏。