目前感觉版本的迭代是真的有点难.

主要是自己的思路并没有很完善...

另一个就是自己太着急了,还没有想好,就应经开始做了...

这样容易烂尾

midi\_read()

作用:读取midi文件,把midi文件转换成对应的二维数组.

实现方式:二维数组分为两层,外面的是note\_list\_ex,里面的一层是note\_list\_in;

上轮为按下+本轮为按下 :连续两轮按下,即为连着,同时按下两个钢琴按键

上轮为按下+本轮为松开 :代表从开了某个钢琴按键

上轮为松开+本轮为松开 :代表连续松开了两个钢琴按键

上轮为松开+本轮为按下 :松开了某个钢琴按键,但是有按下了其他的钢琴按键,即切换按键,开始新的一个音符.

然后根据上面的四种操作,分别对上面四种情况进行编程.

目前bug

假设按下,都按下,松开都松开,这里很规则,那么用下面这个,比较好用.

但是有时候,会出现连续按下三个按键,下一轮松开的时候只松开了两个按键,另一个按键在多轮之后才松开,这个程序就会出问题...

另一个情况是,有空键出现的时候.也就是note on 按下的时候有时间,通常情况下,这里是没有时间的,代表着按下和上次松开没有间隔,但是如果有间隔,就说明,之前有一个空的音符输入...

其实读取midi文件和重新建立midi文件是成对出现的..

关于算法部分:

目前要列出midi中能够代表特征的一些数据.

然后根据这个特征的特点,结合一定的算法.

首先:

Midi中跟音符的序列很重要.

Midi文件的上限是多少.

Midi文件的下限是多少.

midi文件主要集中在哪里.

Midi文件的跨度是多少.

如果这种形式不好处理,可以想一下其他的处理形式,转换成其他的坐标...

或者编写一个流动的以太,让数据进入这个空间,进行流动...