少儿编程

21工作室出品

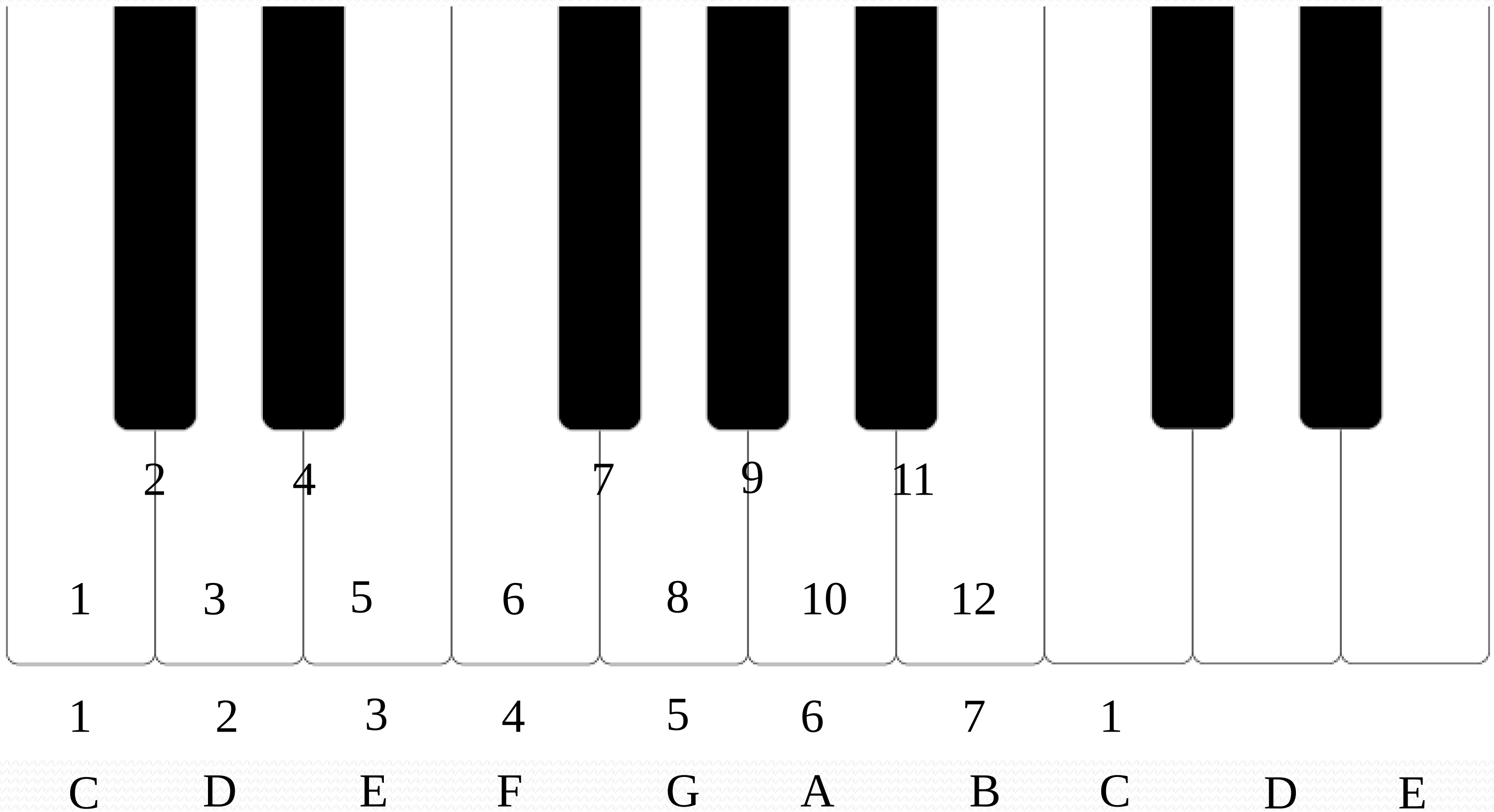
由一个乐理轮盘
到一个上帝随机数

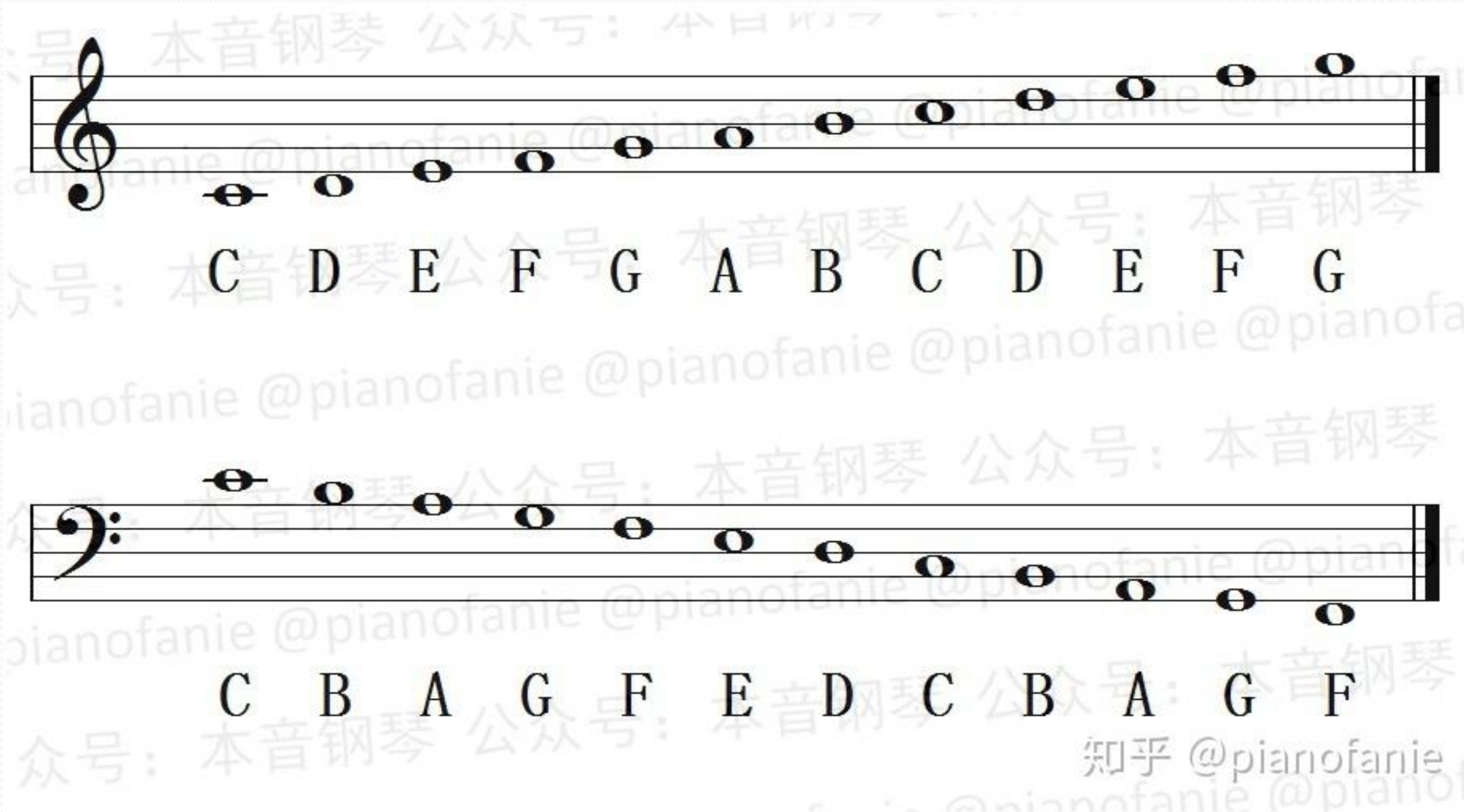
- 你一定有一首或者很多首自己很喜欢的歌曲。
- 在放松的时候你会忍不住哼出声来，甚至有一些神曲在一段时间里好像有魔性似的、不受控制的一直盘绕在你的脑海当中。
- 这就是音乐的魅力，一段好的旋律就像变了程序运转在大脑里带来奇妙的感受。

- 但是你知道为什么所有的音乐都只有自然七声音阶，十二律又是什么吗？

八度和十二律

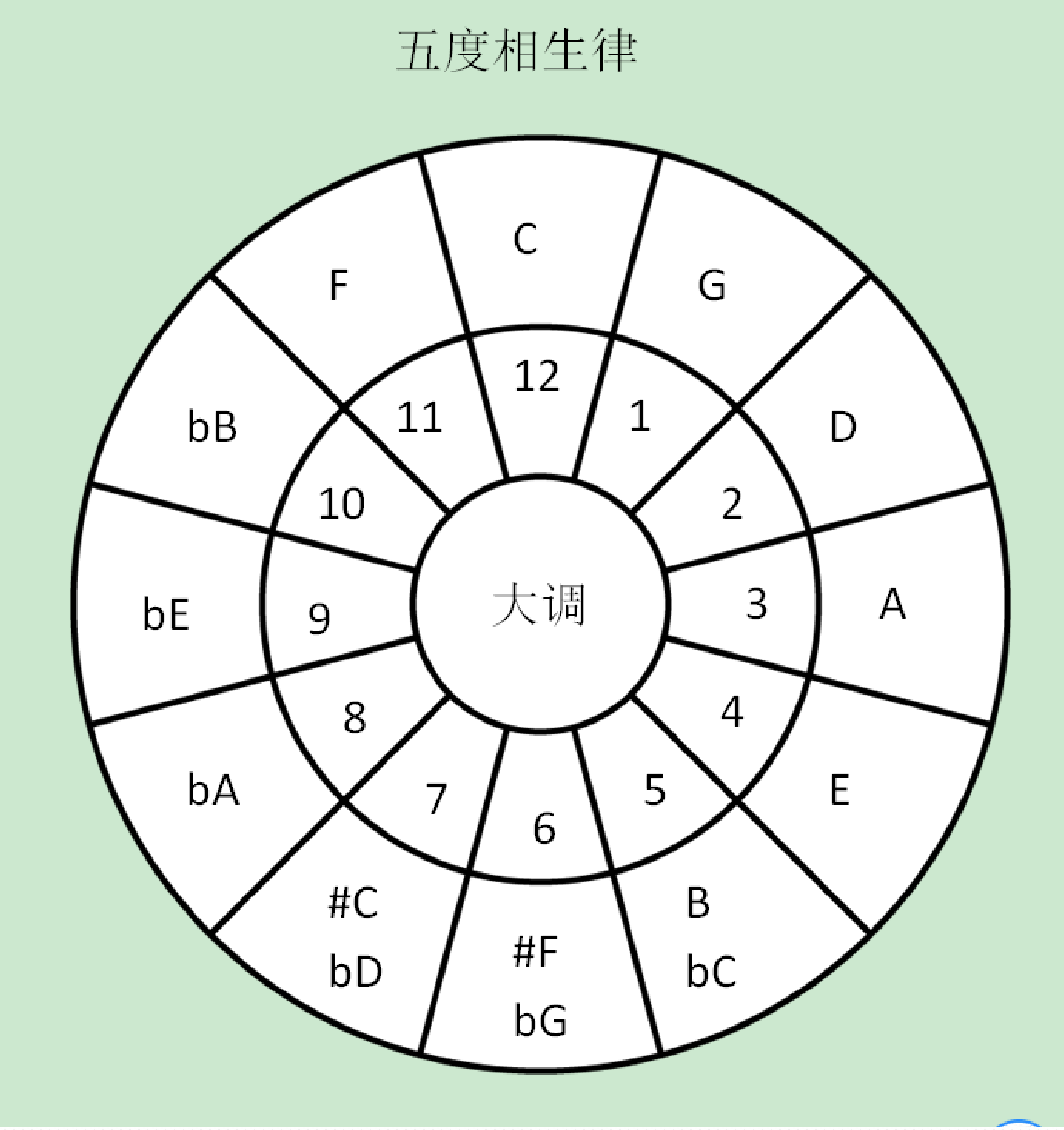
- 在钢琴键盘上




C D E F G A B C D E F G
C B A G F E D C B A G F

音程与比例

- 同一调号，相邻两个音之间有八度的音程。
- 从物理学的角度来讲，构成八度音程的两个音，较高的音的震动频率是较低的音的2倍。
- 音数为6的八度是纯八度。（音程中包含的全音和半音的数量）
- 同样一个自然七声音阶，当演奏者从十二律中选用不同的律来构成而形成不同的调域时，在键盘上表现为不同的选键方式。


五度相生律
C
G
D
A
E
B
bC
#F
bG
#C
bD
bA
bE
bB
F
12
1
2
3
4
5
6
7
8
9
10
11
大调

- 五度循环圈揭示了音与音之间（调与调之间）更深层次的规律：时间每走过12个小时，音就循环一次，并且经历了所有的白键和黑键！

1. 旋转到1点钟，得到：

$$g^1 = c^1 \times \frac{3}{2} = \frac{3}{2}$$

2. 旋转到2点钟，注意这里进入小字二组，得到：

$$d^2 = g^1 \times \frac{3}{2} = \frac{9}{4}$$

向下移动八度，计算出小字一组的d1，得到：

$$d^1 = \frac{d^2}{2} = \frac{9}{8}$$

3. 旋转到3点钟，得到：

$$a^1 = d^1 \times \frac{3}{2} = \frac{27}{16}$$

继续顺时针旋转能推导出一系列的频率比。根据标准音定义为a1=440Hz（第一国际高度），可以通过频率比计算出其他所有音的频率，例如计算c1（中央C）的频率，精确到小数点后两位：

$\because \frac{a^1}{c^1} = \frac{27}{16}$

$\therefore c^1 = \frac{16 \times a^1}{27} = \frac{16 \times 440}{27} \approx 260.74 \approx 260$

这就是五度相生律推导出的中央C的频率！

跟我试一试

1. 找一把吉他，找到1的位置，然后在这根弦被按住的琴柱到吉他底部弦长度的2/3的位置，按住琴柱就能得到5（得到g1音）
2. 同理找到5的那根弦，这次不要按琴柱开始了，直接找到2/3的位置按住琴柱，就是高音2（得到d2音）

1. 再做一个反向的，找到2的那根弦，按住一半的位置，就能得到高音2（得到d1音）

- 这就是刚才数学计算的成果应用。

1. 无升降号的音；

C	D	E	F	G	A	B
1	9/8	81/64	4/3	3/2	27/16	243/128

2.过程

C	D	E	F	G	A	B
1				3/2		
1	9/8			3/2		
1	9/8			3/2	27/16	
1	9/8	81/64		3/2	27/16	
1	9/8	81/64		3/2	27/16	243/128
1	9/8	81/64	4/3(由C向左,乘以2/3再乘以2)	3/2	27/16	243/128

上面是乐理部分、下面是数学部分

- 为什么在12刻度的轮盘里，按步长为5前进，12次就能遍历整个轮盘呢？

- 同余方程 $At \equiv B(modL)$
- 其中A=5，B=1，L=12
- 根据扩展欧几里得算法可得特解t=5（算法附后）

```python
def myExtGCD(a, b):
    """
    a: 模的取值
    b: 想求逆的值
    """
    if b == 0:
        return 1, 0, a
    x, y, gcd = myExtGCD(b, a % b)
    return y, x-a//b*y, gcd


# print(myExtGCD(71, 7)[1] % 71)  # python返回多个值（返回元组），这里我们只显示索引号为1的返回值，并且该值取模71
print(myExtGCD(21, 11)[1])  # 5x+12y=1的特解x=5，退出y=-2
```

打印转盘过程

```python
# pan = int(input("::请输入转盘大小："))
# step = int(input("::请输入步长："))
# # times = int(input("::请输入循环次数："))

pan = 21
step = 11
seed = 0
ls = []

for i in range(pan):
    seed = (seed + step) % pan
    if seed in ls:
        print("列表中元素重复。")
        break
    else:
        ls.append(seed)
print(ls)
# 只要pan和step互质就能遍历。
```

12，5

- [5, 10, 3, 8, 1, 6, 11, 4, 9, 2, 7, 0]
- 5

- 可以看到，按照步长为5的步子迈5次，落在圆盘中的位置就会顺时针播动1.

由同余方程推得随机数

- 有线性同余方法来生成随机数。

- 有如下递归方法：
- seed = (a*seed + c) % m

- 需要满足下面条件：

- 根据Hull-Dobell Theorem，当且仅当：
- 1. c和m互素;
- 2. a-1可被所有m的质因数整除;
- 3. 当m是4的整数倍，a-1也是4的整数倍时，周期为m。所以m一般都设置的很大，以延长周期。

- 当取值:
- m = 28193
  a = 11 * 233 + 1
  c = 100

- 这是产生4800个随机数，看看在80*60的屏幕中分布情况：