

4 – Tesadüfileştirmek

Kendi müziğinizi yaparken müziğinize ilginç eklemeler yapmak için rastgele sayılar kullanmak harika bir yoldur. Sonic Pi, müziğinize rastgelelik katmak için harika bir işleve sahiptir ancak, bunları öğrenmeye başlamadan önce bir gerçekle yüzleşmeniz gerekir: Sonic pi tamamen rastgele değildir. Bu ne demek dediğinizi duyar gibiyiz. Gelin beraber görelim.

-Tekrarlanabilirlik

İşte karşınızda rrand (ranged random) fonksiyonu. “rrand” fonksiyonu size iki sayı arasında bir değer bulmanızı sağlayan oldukça kullanışlı bir fonksiyondur. Haydi rastgele bir nota çalalım:

```
play rrand(50, 95)
```

Vay be, rastgele bir nota çalındı. Bilgisayar 83.7527 notasını seçti ve çaldı. 50 ve 95 arasında güzel bir nota. Ah, bir dakika, gerçekten aynı numara mı rastgele bir biçimde geldi? Tekrar oynatmayı deneyin. Nasıl? Yine mi 83.7527? Bu rastgele olamaz!

Sorunuzun cevabı, rrand gerçekten random değil sahte rastgeledir (pseudo-random). Sonic Pi size rrand fonksiyonuyla tekrarlanan davranışta rastgele benzeri sayılar verecektir. Bu komutu bilgisayarınızda oluşturduğunuz müzikte kullanmanız, her bir bilgisayarda farklı bir müzik yaratmanızı sağlar. Elbette bir rastgelelik fonksiyonu her seferinde 83.7527 sayısını seçse yeterince ilgi çekici olmazdı. Ancak işler görüldüğü gibi değil. Gelin hep beraber bakalım:

```
loop do
  play rrand(50, 95)
  sleep 0.5
end
```

Evet! Sonunda rastgele müziğimiz çalışıyor. Belirli çalışmalar içerisinde rastgele fonksiyonlarla yapılan işlemler müzik içinde rastgele değerlere dönüşür. Ancak ilk çalıştırmadan sonraki çalıştırmalarda pseudo-random’dan dolayı her seferinde aynı ses dizisini verecek ve aynı sesler çıkacaktır. Groundhog Müzik sentezi günü!

-Perili Çanlar

İşte eylemdeki rastgeleliğin güzel bir örneği olan perili çanların rastgele oran ve uyku süreleri:

```
loop do
  sample :perc_bell, rate: (rrand 0.125, 1.5)
  sleep rrand(0.2, 2)
end
```

-Rastgele Ayırımlar

Bir başka eğlenceli rastgelelik örneği ise brleşleri rastgele biçimde ayırmaktır.
:tb303 bunun için harika bir örnektir:

```
use_synth :tb303

loop do
  play 50, release: 0.1, cutoff: rrand(60, 120)
  sleep 0.125
end
```

-Rastgele Kaynaklar

Peki ya rastgele oluşmuş olan müziğimizi değiştirmek istersek ne yapacağız? Bu tamamen use_random_seed ile farklı bir başlangıç seçmeye bağlıdır. Halihazırda kullanılan kaynak değeri 0'dır. Farklı bir kaynak değeri seç ve farklı bir rastgelelik deneyimi yaşa!

```
5.times do
  play rrand(50, 100)
  sleep 0.5
end
```

Bu kod her çalıştırıldığında aynı sırada 5 nota duyulacaktır. Daha farklı sıraya sahip olmak için kaynak değerini değiştir:

```
use_random_seed 40
5.times do
  play rrand(50, 100)
  sleep 0.5
end
```

Bu kod ise ilkinden tamamen farklı sırada notalar çalar ve bu sefer diğer herkes seninle aynı sırada notaları duyar.

-Choose Fonksiyonu

Choose fonksiyonu oluşturduğunuz herhangi bir liste içerisinde bir değer seçmeyi sağlayan bir fonksiyondur.

```
loop do
  play choose([60, 65, 72])
  sleep 1
end
```

-Rrand Fonksiyonu

Bu fonksiyonda daha önce bahsetmiştik ancak yine de tekrar edelim. Rrand fonksiyonu sizin girdiğiniz 2 değer arasında herhangi bir gerçek sayı seçer. Seçilen sayılar hiçbir zaman tam sayı olmaz. İşte `rrand(20, 110)` için bazı örnek değerler:

```
87.5054931640625
86.05255126953125
61.77825927734375
```

-Rrand_i Fonksiyonu

Bazı zamanlarda özellikle rastgele tam sayı kullanmanız gereken durumlarda `rrand_i` sizi kurtarmaya geliyor. Rrand fonksiyonu ile neredeyse aynı. İşte size örnek `rrand_i(20, 110)` değerleri:

```
88
86
62
```

-Rand Fonksiyonu

Rand fonksiyonu sizin için 0 (dahil) ve girdiğiniz maksimum değer (dahil değil) arasında rastgele bir sayı seçer. Eğer herhangi bir sayı girmediyseniz maksimum değeri otomatik olarak 1 alır.

```
loop do
  play 60, amp: rand
  sleep 0.25
end
```

-Rand_i Fonksiyonu

Rand fonksiyonuyla benzer şekilde çalışan `rand_i` fonksiyonu değerler arasından rastgele tam sayı seçer.

-Dice

Zar atışına benzeyen `dice` fonksiyonu ise `rrand_i` gibi girilen değerler arasından rastgele bir tam sayı seçer ancak bu sefer minimum değer 0 yerine 1'dir.

-One_in

Son olarak `one_in` fonksiyonu ise seçilen değerler arasındaki olasılığı bulur. Sayıda 1 ihtimalle True, değilse False sonucunu çıkarır.

Şimdi kodlarınıza biraz rastgelelik ekleyin!