

Gelişmiş Melodi Oluşturma

Bunları daha önce işledik, bu bölümde **chord** ve **scale** metodlarını ayrıca listeler ve değişkenleri nasıl kullanacağımızı öğreneceğiz. Bu bölümün amacı Sonic Pi'nin parametrelerini daha iyi kullanmayı ve değiştirmeyi öğrenmektir.

Akort

Aşağıdaki **chord** fonksiyonunu hatırladınız mı?

```
play (chord :c, :major).choose
# plays a random note of the C major chord (:c, :e or :g)
```

Buna benzer bir **scale**(ölçek) fonksiyonu Sonic Pi'da bulunmaktadır. Scale yalnızca bir akordaki notaları değil, tüm ölçekteki notaları çalar.

```
play (scale :c, :major).choose
# plays a random note of the C major scale (:c, :d, :e, :f, :g, :a or :b)
```

choose metodu bir listeden rastgele eleman seçmeye yarar. Eğer daha planlı bir tarz oluşturmak istiyorsanız Sonic Pi'nin **tick**: metodunu kullanabilirsiniz.

```
live_loop :arp do
  play (scale :e3, :minor_pentatonic).tick, release: 0.1
  sleep 0.125
end
```

Burada, E3 minör pentatonik ölçeğini alıyoruz ve her bir öğeyi geçiyoruz. Bu, ölçek bildiriminin sonuna **tick**: ekleyerek yapılır. Bu akort, canlı döngüde yereldir, böylece her canlı döngü kendi bağımsız akorduna sahip olabilir.

```
live_loop :arp do
  play (scale :e3, :minor_pentatonic).tick, release: 0.1
  sleep 0.125
end

live_loop :arp2 do
  use_synth :dsaw
  play (scale :e2, :minor_pentatonic, num_octaves: 3).tick, release: 0.25
  sleep 0.25
end
```

Rings

Ring sona gelindiğinde başa dönen özel bir listedir. Tick metodunu kullandığımız listelere ring denir. Ayrıca **chord** ve **scale** ring listesi oluşturur. Eğer bir listeyi ring'e dönüştürmek istiyorsanız **.ring** veya **ring** metodunu kullanınız.

```
puts [1, 2, 3, 4].ring #=> (ring 1, 2, 3, 4)
puts (ring 1, 2, 3, 4) #=> (ring 2, 2, 3, 4)
```

```
live_loop :arp do
  play [:c, :e, :d, :f].ring.tick, release: 0.1
  sleep 0.125
end
```

Şimdi daha karışık örneklere bakalım.

```
live_loop :keys do
  use_synth :blade
  play [(chord :Ab, :major7), (chord :Db, :major7), (chord :Bb, :minor7), (chord :Eb, '7')].ring.tick, attack: 0.1, release: 1
  sleep 1
end
```

Artık gerçekten bir melodi yazmaya başlayabilirsiniz.

```
live_loop :keys do
  use_synth :blade
  play [(chord :Ab, :major7), (chord :Db, :major7), (chord :Bb, :minor7), (chord :Eb, "7")].ring.tick, attack: 0.1, release: 1
  sleep 1
end

live_loop :lead do
  use_synth :dsaw
  play (scale :Ab, :major, num_octaves: 3).mirror.tick, attack: 0.05, release: 0.25, cutoff: rrand(80, 110), amp: 0.5
  sleep 0.25
end
```

Değişkenler

Öncelikle değişkenlerin ne olduğuna bakalım.

```
live_loop :melody do
  use_synth :blade
  r = [0.25, 0.25, 0.5, 1].choose
  play (chord :c, :minor).choose, attack: 0, release: r
  sleep r
end
```

Döngü her başa geldiğinde farklı bir değer alan, **r** isimli bir değişken vardır. Değişkenler, içinden istediğini alıp istediğini koyabildiğin bir kutu olarak davranırlar. Kullanımı **variable_name = variable_value** görüldüğü kadar kolaydır.

Hadi örneklerle test edelim! Değişkenlerle test etmek için, **:keys** loop(döngü) metodunu ve **:bass** metodunu kullanarak gösterelim.

```
live_loop :melody do
  use_synth :blade
  r = [0.25, 0.25, 0.5, 1].choose
  play (chord :c, :minor).choose, attack: 0, release: r
  sleep r
end

live_loop :keys do
  use_synth :blade
  play chord(:c, :minor)
  sleep 1
end

live_loop :bass do
  use_synth :fm
  n = (chord :c2, :minor).tick
  3.times do
    play n
    sleep 1
  end
  play n
  sleep 0.5
  play (chord :c2, :minor).tick
  sleep 0.5
end
```

İleri Seviye Akortlar

Önceki örnekte sadece C minörü kullanmıştık. Şimdi ise ring akortlarını ve tüm loopları kullanıyoruz. Kullandığımız tüm looplar aynı akortlar çevresinde oluşuyor. Şimdilik biraz zor görünse de size ilham vereceğine inanıyoruz daha ayrıntılı akortlar öğrenmek için Akortlar Ölçüler ve Teller kısmına bakabilirsiniz.

```
chords = [(chord :C, :minor7), (chord :Ab, :major7), (chord :Eb, :major7), (chord :Bb, "7")].ring
c = chords[0] # take the first chord of the ring and save it to a variable
# this is going to be used in all the live_loops. It will be ticked by the :bass loop

live_loop :melody do
  use_synth :blade
  r = [0.25, 0.25, 0.5, 1].choose
  play c.choose, attack: 0, release: r
  sleep r
end

live_loop :keys do
  use_synth :blade
  play c
  sleep 1
end

live_loop :bass do
  use_synth :fm
  use_octave -2
  3.times do
    play c[0] # play the first note of the chord
    sleep 1
  end
  play c[2] # play the third note of the chord
  sleep 0.5
  play c[1] # # play the second note of the chord
  sleep 0.5
  c = chords.tick
end
```