

Şarkı Yapalım

Hadi bir şarkı besteleyerek Sonic Pi öğrenelim. Giriş bölümünden sonra notaları farklı uzunluklarda çalmayı öğrendik. Çok güzel ama çok daha fazlasını yapabiliriz.

Bu Bölüm

Bu bölümde şarkımızı yaparken önce bir davul ritmi yapıp live_loop kodunun ne yaptığını öğreneceğiz. Sonra şarkımıza bas parça ve melodi ekleyerek şarkımızı tamamlayacağız.

Sonraki Bölüm

Eğer Sonic Pi daha çok öğrenmek istiyorsanız seslerle oynayarak devam edebilirsiniz. Sonraki bölümde ses efektlerini kullanarak daha hoş şarkılar oluşturmayı öğreneceğiz .

Son Bölüm

Anlatacaklarım burada bitmiyor. Son bölümde algoritmalar oluşturarak şarkılarınızın parçalarını oluşturmayı öğreneceğiz.

Davul Ritmi

Hadi basit bir davul ritmi yapmayı öğrenelim. Bu davul ritmini oluştururken iki yeni kod öğreneceğiz : live_loop ve sample.

Boş bir buffer ile başlayın ve bir live_loop oluşturun ve :drums adını koyun. İsterseniz farklı bir ad koyabilirsiniz. Looplarınıza ad vererek ne yaptıklarını hızlı bir şekilde görebilirsiniz. live_loop diğer live-loop lar ile eş zamanda çalan sonsuz döngülerdir. Her live_loop'un içinde en az bir sleep olması gereklidir.

```
live_loop :drums do
  sample :drum_heavy_kick
  sleep 1
end
```

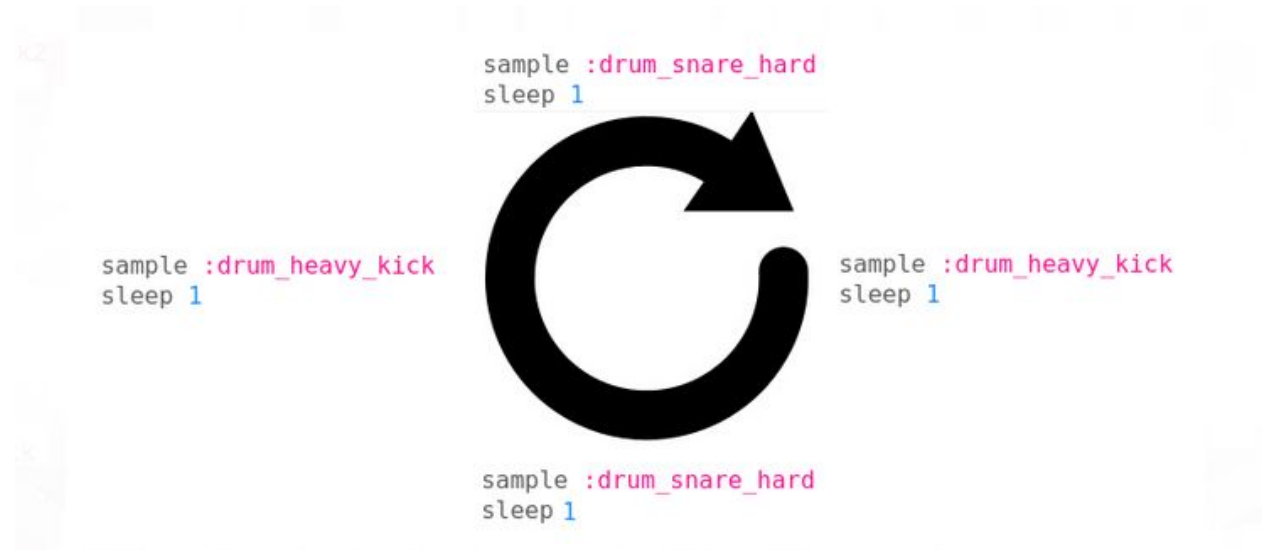
Run'a basın ve ilk live_loop'unuzun güzel bir bas davulu çalışını dinleyin.

Hadi basit bir davul ritmi yapalım. 1. ve 3. vuruşta bas davul 2. ve 4. vuruşta trampet çalacak bir ritim olsun. Şimdi nota çalmak yerine sample çalacağız. “sample :sample_adı” yazmak kadar kolay. Örnek bir davul ritmi:

```
use_bpm 100

live_loop :drums do
  sample :drum_heavy_kick
  sleep 1
  sample :drum_snare_hard
  sleep 1
  sample :drum_heavy_kick
  sleep 1
  sample :drum_snare_hard
  sleep 1
end
```

İşte sağlam bir ritim. “:drums” döngüsü önce bas davul sonra trampet sonra tekrar bas davul sonra tekrar trampet çalıyor. Sonrasında baştan başlayıp döngüye devam ediyor.



Şimdi tempoyu değiştirmeyi deneyin (use_bpm den sonraki sayı) ve sampleları değiştirin. Sample isimlerini yazmaya başladığınız zaman oto-tamamlama özelliği sayesinde farklı samplelara bakabilirsiniz. Farklı samplelar deneyerek neye benzediklerine bak. Fark ettiysen yeni seslerin gelmesi için Stop tuşuna basıp tekrar başlatmana gerek yok sadece Run tuşuna basman yeterli olacak. Bir sonraki döngüde yaptığınız değişiklik müzik durmadan uygulanacak

Şimdi simbal ekleyelim. “:hihat” adlı başka bir live_loop oluşturalım ve simbal samplelarımızı ekleyelim.

```
live_loop :hihat do
  sample :drum_cymbal_closed
  sleep 0.25
end
```

Ama illa bu kadar düz olacak değil. Biraz değişiklik yapalım:

```
live_loop :hihat do
  sample :drum_cymbal_closed
  sleep 0.25
  sample :drum_cymbal_pedal
  sleep 1
end
```

Bu poliritmik simbal döngüsü en başta “yanlış”mış gibi hissettiriyor. Döngü 1 vuruş olacağına 1.25 vuruş uzunluğunda ama güzel çalışıyor. Şu unutmayın “yanlış” yapmaya devam edin ki bilmeden aramadığınız bir şey bulabilirsiniz. Şimdi şarkımız böyle görünüyor.

```
use_bpm 100

live_loop :drums do
  sample :drum_heavy_kick
  sleep 1
  sample :drum_snare_hard
  sleep 1
  sample :drum_heavy_kick
  sleep 1
  sample :drum_snare_hard
  sleep 1
end

live_loop :hihat do
  sample :drum_cymbal_closed
  sleep 0.25
  sample :drum_cymbal_pedal
  sleep 1
end
```

Bas Parça

Şarkımızın bass kısmı için başka bir live_loop döngüsü açalım. Basit ve güzel bir bas parçası besteleyelim.

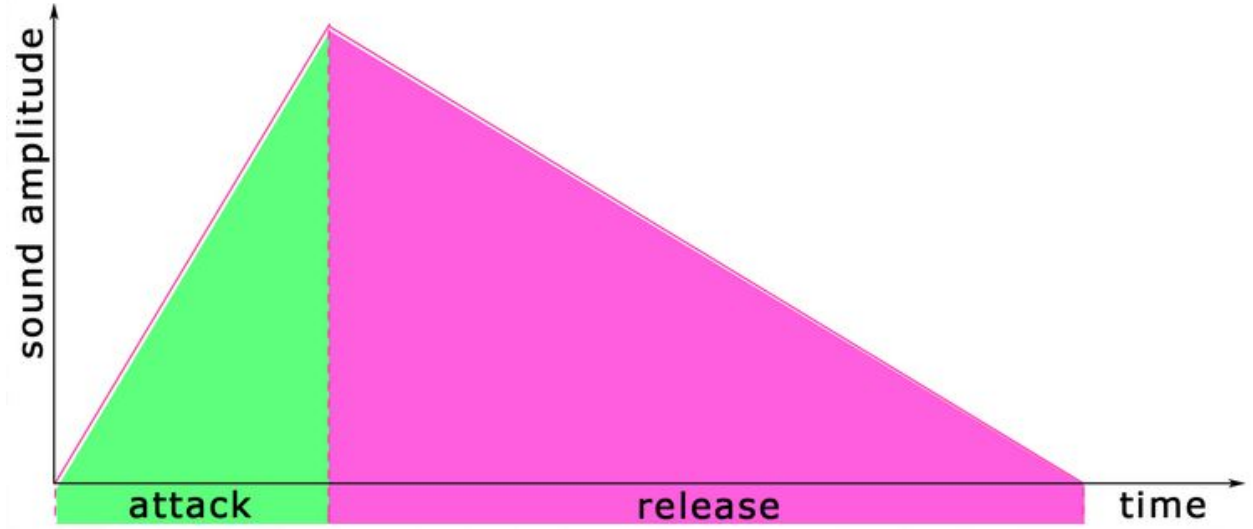
```
live_loop :bass do
  play :c2
  sleep 0.25
  play :c2
  sleep 2
  play :e2
  sleep 0.75
  play :f2
  sleep 1
end
```

Parçamızı daha ilginç yapma zamanı geldi. Bunu yapmak için kullandığımız ses tipini değiştirelim. Sonic Pi'de varsayılan ses tipine “beep” deniyor. Farklı bir ses tipi kullanmak için kullanmak istediğin yerin üstüne “use_synth :synth_adı” yazabilirsin.

Bu örnekte ses tipinin adı fm:

```
live_loop :bass do
  use_synth :fm
  play :c2
  sleep 0.25
  play :c2
  sleep 2
  play :e2
  sleep 0.75
  play :f2
  sleep 1
end
```

Bazı durumlarda çalan seslerin uzunluğunu değiştirmek isteyebilirsiniz. Bunu yapmak için “attack” ve “release” kodlarıyla oynayarak bir sesin zaman içinde nasıl çaldığını kontrol edebilirsiniz.



“attack” ve “release” kodlarını kullanmak örnekteki gibidir. Burada nota 4 vuruşluk.

```
play 60, attack: 1, release: 3
```

Kesik notalar yapmak için “attack” değerini 0 “release” değerini düşük bir değer verebilirsiniz.

```
play :c4, attack: 0, release: 0.2
```

Farklı ses tipleriyle ve nota uzunlukları deneyerek bas parçanı tamamla.
Şimdi parçamız şuna benziyordur.

```

use_bpm 100

live_loop :drums do
  sample :drum_heavy_kick
  sleep 1
  sample :drum_snare_hard
  sleep 1
  sample :drum_heavy_kick
  sleep 1
  sample :drum_snare_hard
  sleep 1
end

live_loop :hihat do
  sample :drum_cymbal_closed
  sleep 0.25
  sample :drum_cymbal_pedal
  sleep 1
end

live_loop :bass do
  use_synth :fm
  play :c2, attack: 0, release: 0.25
  sleep 0.25
  play :c2, attack: 0, release: 0.3
  sleep 2
  play :e2
  sleep 0.75
  play :f2
  sleep 1
end

```

Şimdi şarkımıza bir melodi ekleme zamanı. Bunu yaparken bir sürü “play” ve “sleep” kullanmak yerine bir kısayol kullanacağız: “play_pattern_timed”

Bunu yazmak yerine:

```

play :c2
sleep 0.5
play :d2
sleep 0.25
play :e2
sleep 0.75
play :d2
sleep 0.5

```

Bunu yazabilirsin:

```

play_pattern_timed [:c2, :d2, :e2, :d2], [0.5, 0.25, 0.75, 0.5]

```

İlk liste [:c2, :d2, :e2, :d2] çalınacak notalar ikinci liste [0.5, 0.25, 0.75, 0.5] ise notalar arasındaki boşlukların listesidir.

Melodimiz için “:melody” adlı live_loop açıp melodimizi yapmaya başlayalım. Eğer daha kolay geliyorsa “play_pattern_timed” kullanarak melodini yazabilirsin. Şu an şarkı buna benziyordu.

```
use_bpm 120

live_loop :drums do
  sample :drum_heavy_kick
  sleep 1
  sample :drum_snare_hard
  sleep 1
  sample :drum_heavy_kick
  sleep 1
  sample :drum_snare_hard
  sleep 1
end

live_loop :hihat do
  sample :drum_cymbal_closed
  sleep 0.25
  sample :drum_cymbal_pedal
  sleep 1
end

live_loop :bass do
  use_synth :fm
  play :c2, attack: 0, release: 0.25
  sleep 0.25
  play :c2, attack: 0, release: 0.3
  sleep 2
  play :e2
  sleep 0.75
  play :f2
  sleep 1
end

live_loop :melody do
  play_pattern_timed [:c4, :e4, :f4, :g4, :f4, :e4, :f4, :g4, :f4, :e4, :f4], [0.25, 0.25, 0.25, 1.5, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25], attack: 0, release: 0.2
end
```

Güzel bir başlangıç yaptık. Şimdi istediğin şarkıyı besteleyebilirsin.