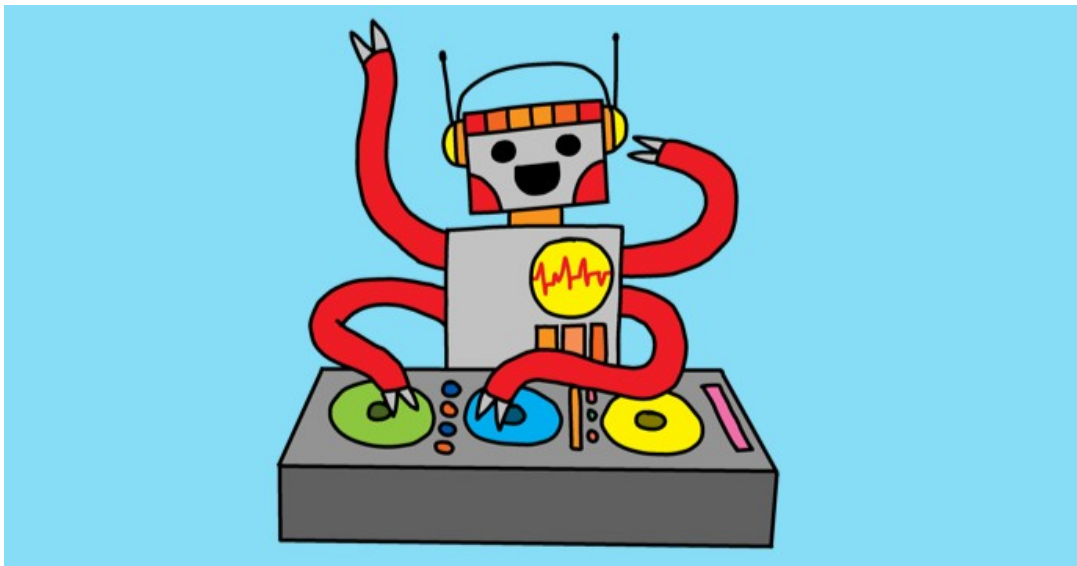


π))) Sonic Pi

Live DJ



Introductie

In dit project ga je leren om een live muziekoptreden te coderen, waarin je kunt toevoegen en veranderen zonder de muziek te stoppen.

Niveau: Uitdagend.

Drums

Laten we beginnen met het creëren van een simpele drum loop.

- Begin met het creëren van een `live_loop` genaamd `:drums`.

```
live_loop :drums do  
  
end
```

Alle code toegevoegd aan een `live_loop` wordt herhaald tot 'Stop' is ingedrukt.

- Voeg afwisselde basdrum en snaardrum toe die ieder een tel spelen.

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

- Druk op 'Run' om je drum loop te testen.
- Als je drum loop te snel of te langzaam is dan kun je de beats per minuut (bpm) aanpassen.

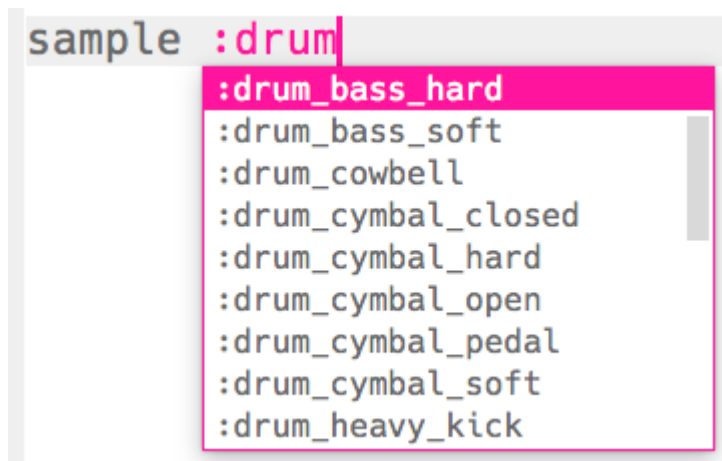
```
use_bpm 65  
  
live_loop :drums do  
  sample :drum_heavy_kick  
  sleep 1  
  sample :sn_dolf  
  sleep 1  
end
```

Je zult je `live_loop` moeten stoppen en starten om de snelheid te verhogen of te verlagen.

Uitdaging: De drum loop veranderen

Kun je andere samples in je drum loop gebruiken?

Om te zien welke samples beschikbaar zijn kun je naar jumpton.cc/sonic-pi-samples gaan, of typ gewoon `sample :drum` en kies uit de lijst die verschijnt.



Een sample toevoegen

Laten we een herhalende sample toevoegen naast de basis drum loop.

- Om een sample te spelen tegelijk met je drums creëer je nog een `live_loop` genaamd `:sample`.

```
use_bpm 65

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

live_loop :sample do
end
```

- Voeg de sample `:loop_compus` toe die iedere 8 tellen speelt.

```
live_loop :sample do
  sample :loop_compus
  sleep 8
end
```

- Als je je sample test dan zul je ontdekken dat het **helemaal niet bij de drums past!**
- Het eerste wat je moet doen is je sample synchroon laten lopen met de drum beat.

```
live_loop :sample do
  sync :drums
  sample :loop_compus
  sleep 8
end
```

- Dit klinkt nog steeds niet juist! Voeg code toe die de tijdsduur van de sample afdrukt:

```
10 live_loop :sample do
11   sync :drums
12   puts sample_duration(:loop_compus)
13   sample :loop_compus
14   sleep 8
15 end
```

- Als je in de log (uitvoer van Sonic Pi) kijkt, zie je dat ondanks dat de sample iedere 8 tellen herhaald wordt de sample geen 8 tellen duurt.

Log

```
{run: 29, time: 0.0}
└─ cue :sample
   sync :drums

{run: 29, time: 0.0}
└─ synced :drums (Run 29)
   7.027025699168555
   sample "/Applications/Sonic Pi.app/etc/samples",
        "loop_compus.flac"
```

(Je kunt nu de code die de tijdsduur van de sample afdrukt weghalen.)

- Om je sample aan te passen aan de drums moet je de sample **uitrekken** zodat deze ook precies 8 tellen duurt.

```
live_loop :sample do
  sync :drums
  sample :loop_compus, beat_stretch: 8
  sleep 8
end
```

- Test je code door weer op 'Run' te drukken – **je hoeft de muziek niet te stoppen en herstarten!** Je zou nu moeten horen dat de sample tegelijk met je drum beat speelt.

Uitdaging: De sample veranderen

Kun je de gebruikte sample veranderen?

Om te zien welke loop samples beschikbaar zijn kun je naar jumpton.cc/sonic-pi-samples gaan, of typ gewoon `sample :loop` en kies uit de lijst die verschijnt.

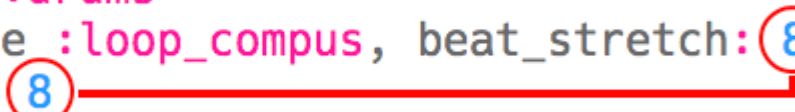
```
live_loop :sample do
  sync :drums
  sample :loop, beat_stretch: 8
  sleep 8 :loop_amen
end
:loop_amen_full
:loop_breakbeat
:loop_compus
```

Je moet misschien ook de getallen in je code aanpassen voor andere samples. Je kunt de volgende code gebruiken om de tijdsduur van een sample uit te vinden:

```
puts sample_duration(:sample_name)
```

Om een sample te herhalen zonder gat, **zorg dat beide getallen overeen komen.**

```
live_loop :sample do
  sync :drums
  sample :loop_compus, beat_stretch: 8
  sleep 8
end
```



Bas toevoegen

Laten we nu wat basnoten toevoegen aan onze muziek.

- Begin met het creëren van een nieuwe `live_loop` genaamd `:bass`. Deze nieuwe loop moet ook synchroon lopen met de drums.

```
sync :drums
sample :loop_compus,
sleep 8
end

live_loop :bass do
  sync :drums
end
```

- Voeg code toe om een enkele noot iedere 8 tellen te spelen. De gespeelde noot gebruikt de `chipbass synth` (afkorting van `synthesizer`).

```
live_loop :bass do
  sync :drums
  use_synth :chipbass
  play 36
  sleep 8
end
```

- Druk op 'Run' (niet nodig om je muziek te stoppen en herstarten). Je zou een noot moeten horen die iedere 8 tellen gespeeld wordt.
- Een **akkoord** is een groep noten die samen gespeeld worden.

In plaats van steeds dezelfde noot te spelen iedere 8 tellen, kun je ook een willekeurige noot kiezen uit een **akkoord**. In dit geval is het akkoord **C Mineur**.

```
live_loop :bass do
  sync :drums
  use_synth :chipbass
  play chord(:c, :minor).choose
  sleep 8
end
```

- 'Middelste' C is eigenlijk :c4. Om lagere basnoten te spelen moet een getal lager dan 4 achter de akkoordnaam gezet worden.

```
live_loop :bass do
  sync :drums
  use_synth :chipbass
  play chord(:c2, :minor).choose
  sleep 8
end
```

- Gebruik een sustain om te kiezen hoeveel tellen de noot wordt aangehouden.

```
live_loop :bass do
  sync :drums
  use_synth :chipbass
  play chord(:c2, :minor).choose, sustain: 7
  sleep 8
end
```

- Je kunt ook amp gebruiken om het volume van de bas te kiezen. Een getal kleiner dan 1 betekent zachter en een getal groter dan 1 luider.

```
live_loop :bass do
  sync :drums
  use_synth :chipbass
  play chord(:c2, :minor).choose, sustain: 7, amp: 0.7
  sleep 8
end
```

- Je kunt ook een (luidere) sample toevoegen die aan het begin van iedere noot speelt.

```
live_loop :bass do
  sync :drums
  use_synth :chipbass
  sample :bd_sone, amp: 3
  play chord(:c2, :minor).choose, sustain: 7, amp: 0.5
  sleep 8
end
```


- Druk op 'Run' om je code te testen. Het is niet nodig je muziek te stoppen en herstarten.

Uitdaging: De bas veranderen

Kun je de basnoten in je muziek aanpassen? Je zou kunnen veranderen:

- De naam van het gespeelde akkoord, bijvoorbeeld f2 in plaats van c2
- Het type akkoord, bijvoorbeeld :major in plaats van :minor
- De gebruikte synth
- De gebruikte sample

```
sync :drums
use_synth |
sample :beep
play chord :blade,
sleep 8
end
```

```
use_synth :chipbass
sample |, amp: 3
play chord :ambi_choir,
sleep 8
end
```

Geluidseffecten toevoegen

Laten we tenslotte nog enkele geluidseffecten aan onze muziek toevoegen.

- Voeg nog een `live_loop` genaamd `:effects` toe die synchroon loopt met de drums.

```
play chord(:C2, :m3)
sleep 8
end

live_loop :effects do
  sync :drums
end
```

- Voeg de volgende code toe die de `:elec_blip2` sample iedere 2 tellen speelt.

```
live_loop :effects do
  sync :drums
  sample :elec_blip2
  sleep 2
end
```

- Druk op 'Run' om je code te testen (niet nodig je muziek te stoppen en herstarten). Je zou een piep effect iedere 2 tellen moeten horen.
- In plaats van iedere keer hetzelfde effect te spelen kun je ook willekeurig uit een list van 2 effecten kiezen.

```
live_loop :effects do
  sync :drums
  sample choose([:elec_blip2, :elec_twip])
  sleep 2
end
```

- Druk op 'Run' om je willekeurige effecten te testen (niet nodig je muziek te stoppen en herstarten).

Uitdaging: De effecten veranderen

Kun je nog meer geluidseffecten aan je muziek toevoegen?

```
live_loop :effects do
  sync :drums
  sample choose( [:elec_blip2, :elec_twip, :elec_beep, :elec_ping] )
  sleep 2
end
```

Uitdaging: Toon je DJ-kunsten

Gebruik alles wat je geleerd hebt om als DJ voor je vrienden op te treden! Bedenk dat je je muziek kunt uitbreiden, maar ook noten en samples veranderen zonder de muziek te stoppen.