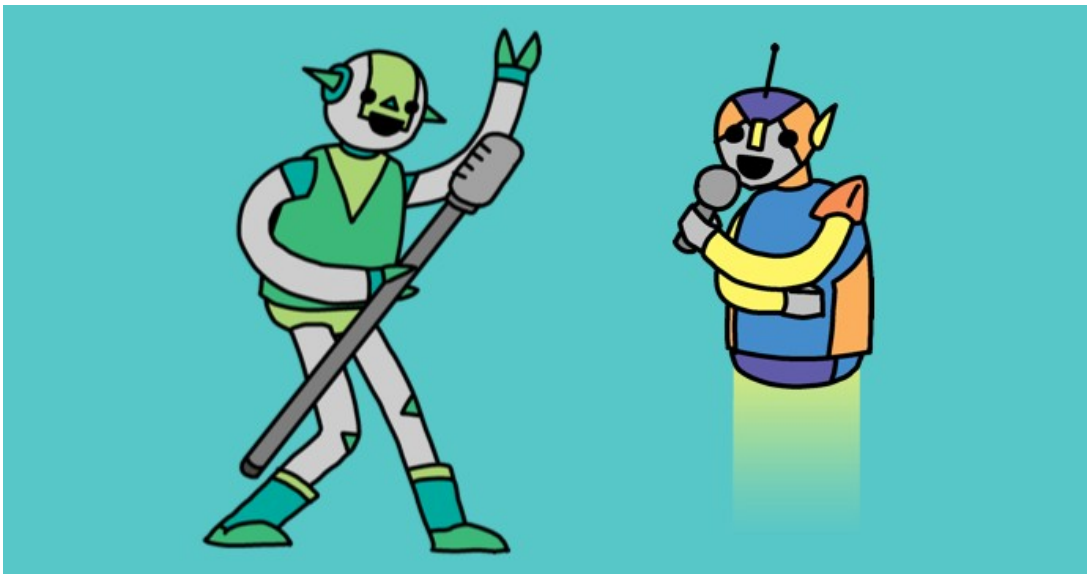
Sonic Pi

Muzikale canon



Introductie

In dit project ga je een canon bouwen waarin verschillende instrumenten dezelfde melodie spelen maar op verschillende tijdstippen starten.

Niveau: Uitdagend.

De melodie

In dit project gebruiken we een melodie die we je al vooraf geven.

- Kies een buffer in Sonic Pi en typ de volgende code in.

```
2.times do
  play_pattern_timed [:c, :d, :e, :c], [0.5]
end

2.times do
  play_pattern_timed [:e, :f, :g], [0.5, 0.5, 1]
end

2.times do
  play_pattern_timed [:g, :a, :g, :f], [0.25]
  play_pattern_timed [:e, :c], [0.5]
end

2.times do
  play_pattern_timed [:c, :g3, :c], [0.5, 0.5, 1]
end
```

Voer deze code uit. Herken je de melodie?

- Laten we deze melodie een naam geven zodat we de melodie kunnen afspelen als we willen.

```
define :fj do
  2.times do
    play_pattern_timed [:c, :d, :e, :c], [0.5]
  end

  2.times do
    play_pattern_timed [:e, :f, :g], [0.5, 0.5, 1]
  end

  2.times do
    play_pattern_timed [:g, :a, :g, :f], [0.25]
    play_pattern_timed [:e, :c], [0.5]
  end

  2.times do
    play_pattern_timed [:c, :g3, :c], [0.5, 0.5, 1]
  end
end
```

- Als je deze code uitvoert, gebeurt er niets. Je moet Sonic Pi nog 'fj' laten spelen.

  Voeg de volgende regel toe aan het einde van je code.

```
  2.times do
    play_pattern_timed [:c, :g3, :c], [0.5, 0.5, 1]
  end
end

fj
```

- Probeer de melodie te spelen met twee verschillende instrumenten.

```
  2.times do
    play_pattern_timed [:c, :g3, :c], [0.5, 0.5, 1]
  end
end

fj

use_synth :piano
fj
```

  De instrumenten spelen nu na elkaar.

Samenspel

Laten we nu eens twee instrumenten laten samenwerken om de melodie te spelen.

- We willen niet dat het tweede versie moet wachten tot de eerste klaar is, dus we moeten Sonic Pi vertellen dat het niet moet wachten. We doen dit door iedere versie te laten spelen in een 'thread'.

```
in_thread do
  fj
end

in_thread do
  use_synth :piano
  fj
end
```

In computertaal noemen we acties die op hetzelfde moment uitgevoerd worden “concurrency”.

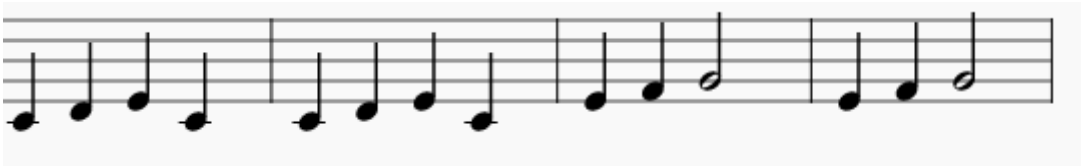
- Voer je code uit en kijk of je twee instrumenten kunt horen.
- Kijk naar de uitvoer en je zult zien dat dezelfde noten door beide instrumenten op hetzelfde tijdstip worden gespeeld:

```
{run: 3, time: 0.0}
  └─ synth :piano, {note: 60.0}

{run: 3, time: 0.0}
  └─ synth :beep, {note: 60.0}

{run: 3, time: 0.5}
  └─ synth :beep, {note: 62.0}

{run: 3, time: 0.5}
  └─ synth :piano, {note: 62.0}

{run: 3, time: 1.0}
  └─ synth :piano, {note: 64.0}

{run: 3, time: 1.0}
  └─ synth :beep, {note: 64.0}

{run: 3, time: 1.5}
  └─ synth :piano, {note: 60.0}

{run: 3, time: 1.5}
  └─ synth :beep, {note: 60.0}
```

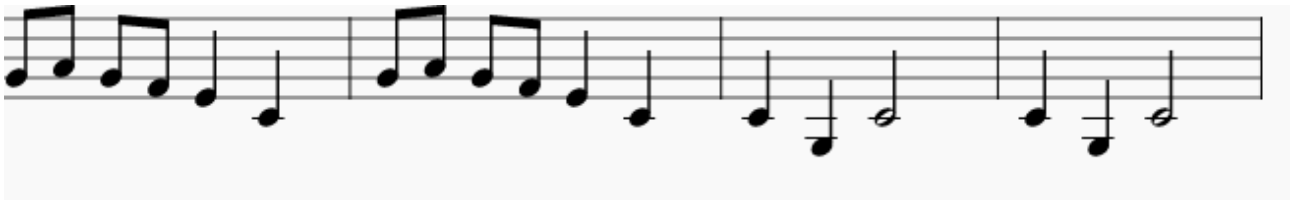
Ieder tijdstip is met een andere kleur aangegeven.

- Laten we eens naar de muziek van dit liedje kijken.

  Dit zijn de eerste vier maten:



Dit zijn de laatste vier maten:



Voer je Sonic Pi project nog een keer uit en volg wat er gebeurt.

- Vader Jacob is een zogenaamde canon. Deze is zo ontworpen dat het goed klinkt als meerdere versies op verschillende tijdstippen starten. Misschien heb je tijdens

muziekles op school wel eens meegedaan aan een canon.

Laten we eens een 'sleep' toevoegen voor de piano begint te spelen.

```
in_thread do
  fj
end

in_thread do
  sleep 4
  use_synth :piano
  fj
end
```

Hoe klinkt het nu?

- Kijk naar de uitvoer van Sonic Pi. Kun je zien wanneer de piano begint te spelen? En wanneer het eerste instrument stopt met spelen?

```
{run: 2, time: 3.0}
 └─ synth :beep, {note: 64.0}

{run: 2, time: 3.5}
 └─ synth :beep, {note: 60.0}

{run: 2, time: 4.0}
 └─ synth :beep, {note: 64.0}

{run: 2, time: 4.0}
 └─ synth :piano, {note: 60.0}

{run: 2, time: 4.5}
 └─ synth :piano, {note: 62.0}

{run: 2, time: 4.5}
 └─ synth :beep, {note: 65.0}

{run: 2, time: 5.0}
 └─ synth :beep, {note: 67.0}

{run: 2, time: 5.0}
 └─ synth :piano, {note: 64.0}
```

Dit is maar een klein stukje. Kijk naar de Sonic Pi uitvoer om het hele stuk te zien.

Uitdaging: Meer instrumenten

Kun je twee extra instrumenten (synthesizers) toevoegen die Vader Jacob spelen zodat ieder 4 maten extra wacht?