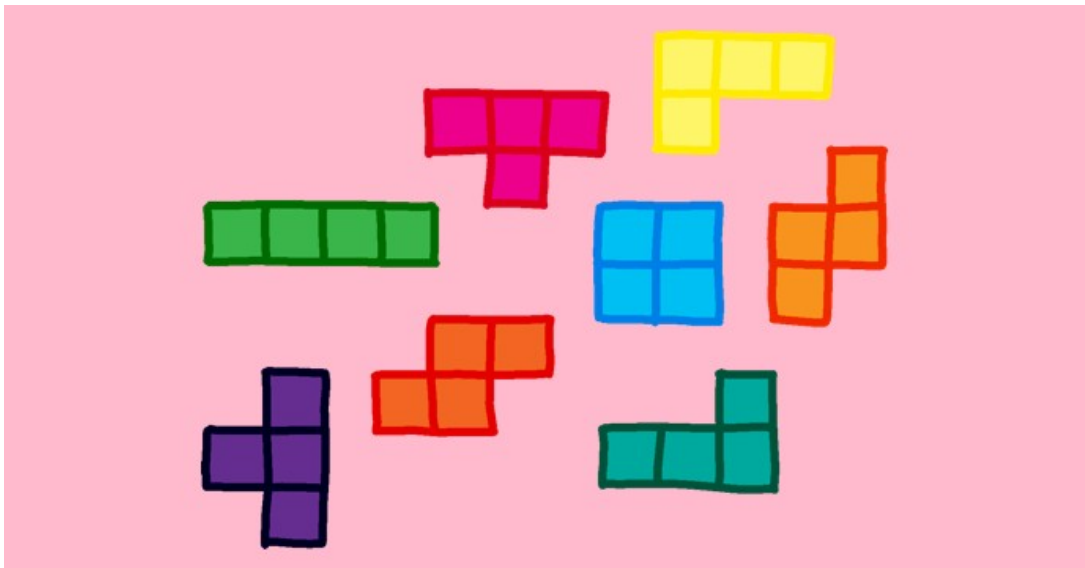


π))) Sonic Pi

Codeer het Tetris thema



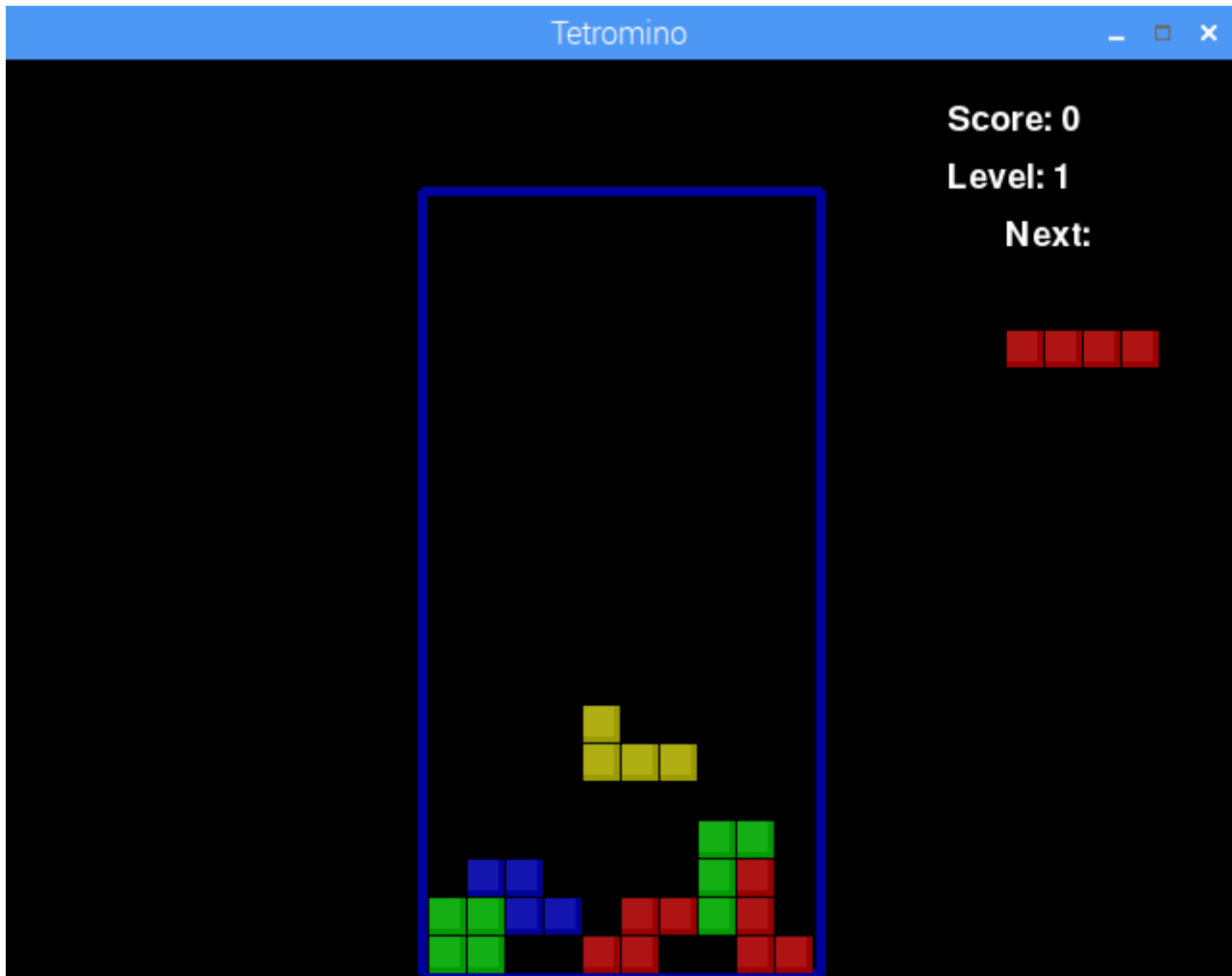
Introductie

In dit project ga je leren om muzieknoten in Sonic Pi te programmeren zodat je bladmuziek in code kunt omzetten. Je creëert je eigen mix van het muzikale thema van Tetris.

Niveau: Uitdagend.

Tetris is een populair computerspel gecreëerd in 1984 door de Russische programmeur Alexey Pajitnov. Veel versies van Tetris gebruiken de muziek 'Korobeiniki', een Russische volksdans.

Als je een Raspberry Pi computer gebruikt dan kun je de muziek horen door het Tetrimono spel te spelen. Je kunt het vinden door 'Games' te kiezen in het menu en dan 'Python Games'. Speel niet te lang! Je moet aan het coderen.



De vormen van de Tetris-stukjes worden tetrominoes genoemd – de 2D vormen die gemaakt kunnen worden met 4 vierkanten.

Codeer het begin

Laten we het eerste deel van de muziek coderen in Sonic Pi.

- Om te beginnen kiezen we een snelheid en een synthesizer voor de muziek. De normale snelheid is 60 tellen per minuut (beats per minute, bpm), maar dat is niet snel genoeg voor dit stuk.

Kies een lege Buffer in Sonic Pi en voeg deze code toe:

```
use_bpm 120  
use_synth :dpulse
```

- Hier is de eerste maat van Korobeiniki:



Het toont onderaan de naam van de noot in het rood en de lengte van de noot bovenaan in het groen.

Muzieknoten hebben namen met een letter van A tot en met G. Om meer noten te krijgen laat je de letter volgen door een cijfer die de octaaf aangeeft (series van lagere of hogere noten). Iedere octaaf start bij C.

In Sonic Pi is de standaard octaaf 4, b3 betekent noot b van een octaaf lager.

Sonic Pi staat je toe letternamen te gebruiken in plaats van getallen. Dit is nuttig als je met notenbalken werkt.

Codeer de eerst 3 noten van Korobeiniki in Sonic Pi:

```
use_bpm 120
use_synth :dpulse
```

```
play :e
sleep 1
play :b3
sleep 0.5
play :c
sleep 0.5
```

- Dat werkt maar vergt nogal wat typewerk. Er is een snellere manier om langere stukken muziek te programmeren: `play_pattern`.

`play_pattern` staat je toe meerdere noten in een regel te programmeren.

Vervang je code om `play_pattern` te gebruiken voor het spelen van de eerste maat:

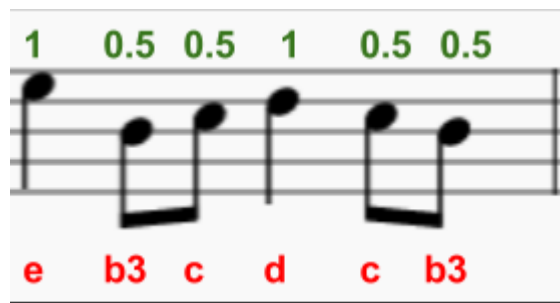
```
use_bpm 120
use_synth :dpulse
```

```
play :e
sleep 1
play :b3
sleep 0.5
play :c
sleep 0.5
```

```
play_pattern [:e, :b3, :c, :d, :c, :b3]
```

- Misschien is het je opgevallen dat de noten niet allemaal dezelfde lengte hebben. Dat is prima. Als `play_pattern` verandert in `play_pattern_timed` kun je zeggen hoe lang iedere noot moet duren.

De getallen in het groen tonen hoeveel tellen iedere noot duurt.



(Als je notenbalken kunt lezen: dit stuk is in 4 kwartsmaat en een kwartnoot duurt 1 tel, een achtste noot duurt een halve tel en een halve noot duurt 2 tellen.)

play_pattern_timed krijgt een lijst noten gevolgd door een lijst tijdsduren.

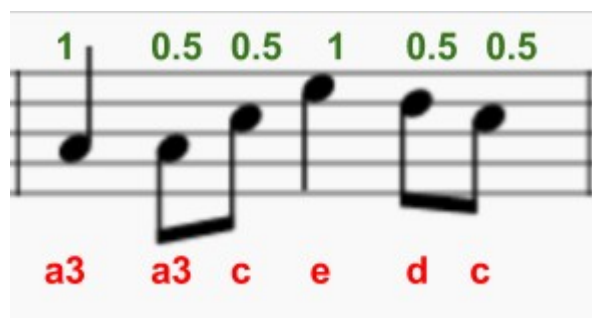
Verander je play_pattern code zodat het er zo uitziet:

```
play_pattern_timed [:e, :b3, :c, :d, :c, :b3], [1, 0.5, 0.5, 1, 0.5, 0.5]
```

- Als je te weinig tijdsduren geeft dan herhaalt Sonic Pi ze. Deze maat herhaalt het tijdspatroon 1, 0.5, 0.5 (kwartnoot, achtste noot, achtste noot), dus je kunt je code veranderen in:

```
play_pattern_timed [:e, :b3, :c, :d, :c, :b3], [1, 0.5, 0.5]
```

- Voeg de volgende maat aan de muziek toe; de tijden zijn hetzelfde voor deze maat.



```
play_pattern_timed [:a3, :a3, :c, :e, :d, :c], [1, 0.5, 0.5]
```

Beginnt de melodie bekend te klinken?

Uitdaging: Maak de melodie af

Hier is de muziek voor de volgende twee maten.



Kun je `play_pattern_timed` gebruiken om het te programmeren?

Sneller!

Russische volksliedjes worden vaak sneller en sneller. Het Tetris thema doet dit ook als je in hogere levels van het spel komt.

Laten we je melodie sneller en sneller laten gaan.

- Je code zou er nu als volgt uit moeten zien:

```
use_bpm 120
use_synth :dpulse

play_pattern_timed [:e, :b3, :c, :d, :c, :b3], [1, 0.5, 0.5]
play_pattern_timed [:a3, :a3, :c, :e, :d, :c], [1, 0.5, 0.5]
play_pattern_timed [:b3, :c, :d, :e], [1.5, 0.5, 1, 1]
play_pattern_timed [:c, :a3, :a3], [1, 1, 2]
```

- Laten we de muziek eerst in een loop plaatsen zodat het meerdere keren speelt:

```
3.times do
  play_pattern_timed [:e, :b3, :c, :d, :c, :b3], [1, 0.5, 0.5]
  play_pattern_timed [:a3, :a3, :c, :e, :d, :c], [1, 0.5, 0.5]
  play_pattern_timed [:b3, :c, :d, :e], [1.5, 0.5, 1, 1]
  play_pattern_timed [:c, :a3, :a3], [1, 1, 2]
end
```

- We kunnen de variabele `current_bpm` gebruiken om uit te vinden waar de bpm op dit moment op staat.

Voeg een regel toe waarin de bpm verhoogd wordt iedere keer dat de loop herhaald wordt:

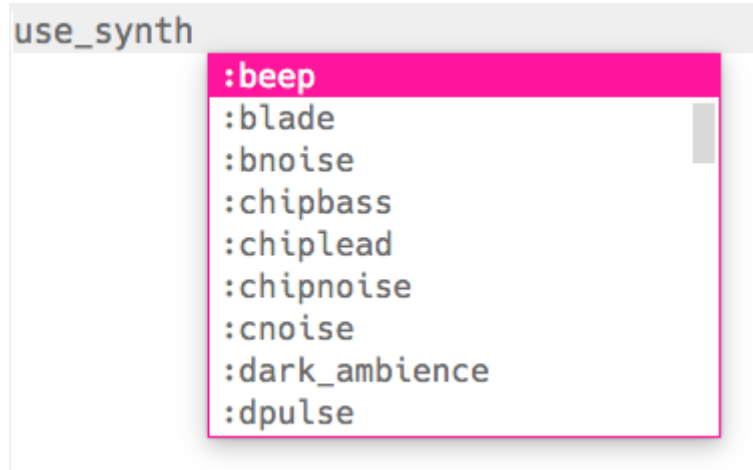
```
3.times do
  play_pattern_timed [:e, :b3, :c, :d, :c, :b3], [1, 0.5, 0.5]
  play_pattern_timed [:a3, :a3, :c, :e, :d, :c], [1, 0.5, 0.5]
  play_pattern_timed [:b3, :c, :d, :e], [1.5, 0.5, 1, 1]
  play_pattern_timed [:c, :a3, :a3], [1, 1, 2]
  use_bpm current_bpm + 40
end
```

Probeer het aantal herhalingen te verhogen om sneller en sneller te gaan!

Uitdaging: Mix de melodie

Probeer de manier waarop de melodie klinkt te veranderen door een andere synthesizer te gebruiken.

Je kunt een lijst met beschikbare synthesizers te zien krijgen door in `use_synth` `[space]` te typen en te kiezen uit de getoonde lijst.



Uitdaging: Meer van de melodie

Kun je het volgende deel van de melodie programmeren:



Sonic Pi gebruikt kleine letter 'r' voor een rust (een pauze) en je kunt dit net als een noot gebruiken.



Uitdaging: Meer muziek

Kun je wat je geleerd hebt gebruiken om meer muziek te programmeren? Je kunt online naar bladmuziek zoeken. Sommige bladmuziek geeft de letternamen voor de noten wat het wat makkelijker maakt.