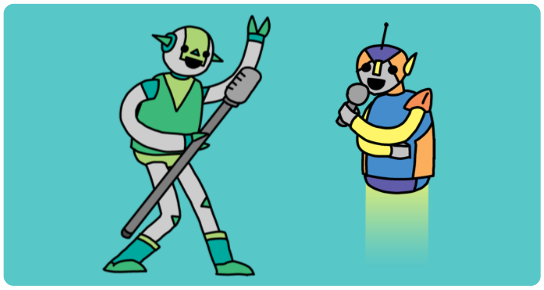
Projects

Musical Round

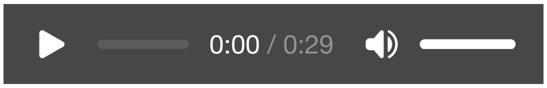
Experiment with synths and timing to create a musical round



Step 1 Introduction

In this project you will create a musical round where different instruments play the same tune but start at different times.

Click the play button below to hear how your musical round will sound:



Do you recongnise the tune?

Additional information for club leaders

If you need to print this project, please use the Printer friendly version (https://projects.raspberrypi.org/en/projects/musical-round/print).

ℹ Club leader notes

Introduction:

In this project, children will program a musical round (Frere Jacques) to learn how to play sounds concurrently and program music using letter names.

Resources

The ‘Project Materials’ link for this project contains the following resources:

Project Resources

- frerejacques.txt

Club leader Resources

You can find a completed version of this project by clicking the ‘Project Materials’ link for this project, which contains:

- frerejacquesround.txt
- frerejacquesround.mp3

Learning Objectives

- Use threads to play multiple sounds concurrently.

This project covers elements from the following strands of the Raspberry Pi Digital Making Curriculum (http://rpf.io/curriculum):

- Use basic programming constructs to create simple programs. (https://www.raspberrypi.org/curriculum/programming/creator)

Challenges

- “More instruments” - Add two more instruments to add more parts to the round.
- “More concurrency” - use the skills learned to program a new piece of music that uses concurrency.

Frequently Asked Questions

Project materials

Project resources

- Downloadable starter Sonic Pi project (https://projects-static.raspberrypi.org/projects/musical-round/4bb1c5c4761dcc63f83da5305f102fcb5d184b05/en/resources/frerejacques.txt)

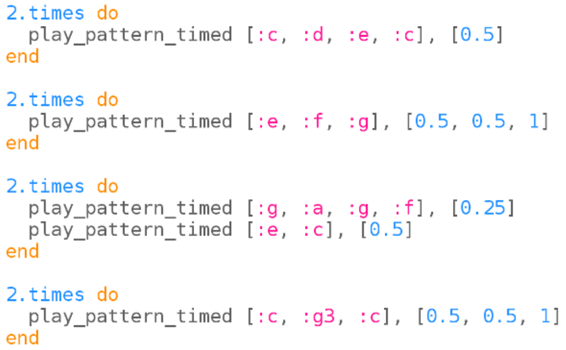
Club leader resources

- Downloadable completed Sonic Pi project (https://projects-static.raspberrypi.org/projects/musical-round/4bb1c5c4761dcc63f83da5305f102fcb5d184b05/en/resources/frerejacquesround.txt)
- Downloadable completed project mp3 file (https://projects-static.raspberrypi.org/projects/musical-round/4bb1c5c4761dcc63f83da5305f102fcb5d184b05/en/resources/frerejacquesround.mp3)

Step 2 The tune

In the last project you used Sonic Pi to program music using letter names. This time we've provided the music for you.

- Choose a buffer in Sonic Pi and Load `frerejacques.txt`. Check with your Club Leader if you don't know where the file is.

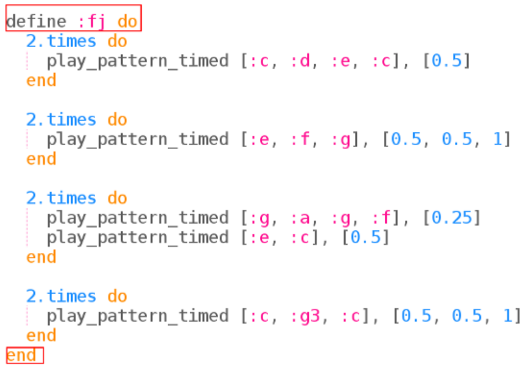


```
2.times do
  play_pattern_timed [:c, :d, :e, :c], [0.5]
end

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
```

Run it. Do you recognise the tune?

0:00 / 0:16

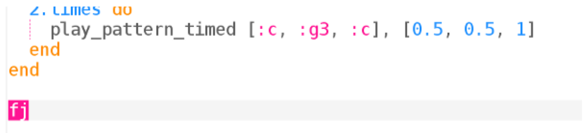
- Let's give our tune a name so that we can play it when we want to.



```
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
```

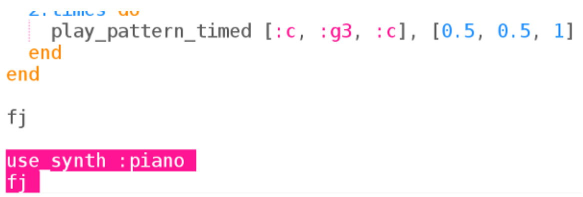
- Now nothing will happen if you run your code. You need to tell Sonic Pi to play 'fj'.

  Add the following line to end bottom of your code:



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

fj
```

- Try playing the tune with two different instruments:



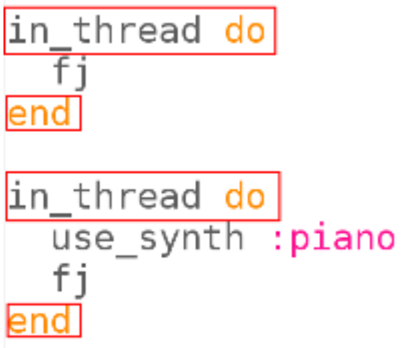
```
  2.times do
    play_pattern_timed [:c, :g3, :c], [0.5, 0.5, 1]
  end
end

fj

use_synth :piano
fj
```

The instruments play one after the other.

Step 3 Concurrency

Now let’s get two instruments working together to play the tune.

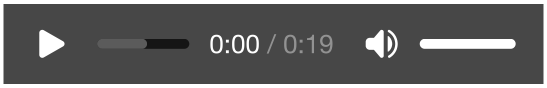
- We don’t want the second version to wait until the first has finished so we’ll need to tell Sonic Pi that it doesn’t need to wait. We do this by running each version inside a ‘thread’.



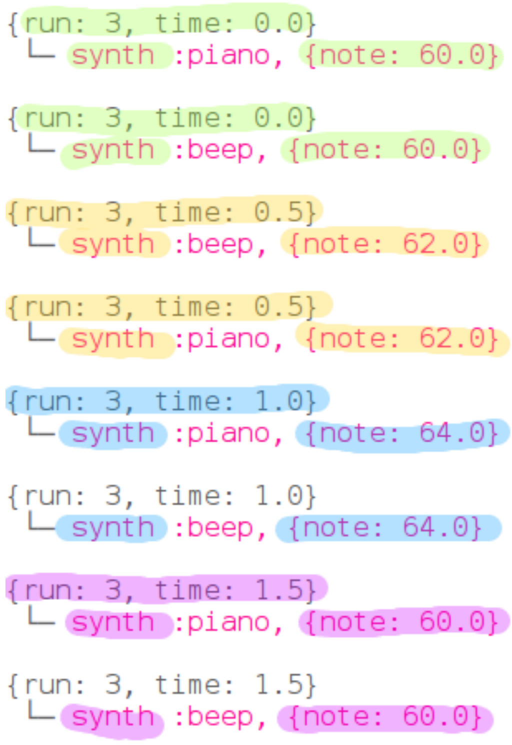
```
in_thread do
  fj
end

in_thread do
  use_synth :piano
  fj
end
```

In computing we call things happening at the same time ‘concurrency’.

- Run your code and see if you can hear two instruments.



- Look at the output and you will see the same notes being played by both instruments at the same time:

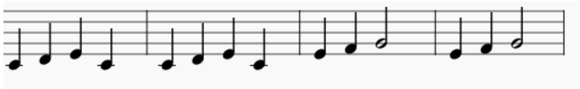


```
{run: 3, time: 0.0}
 └─ synth :piano, {note: 60.0}

{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}
```

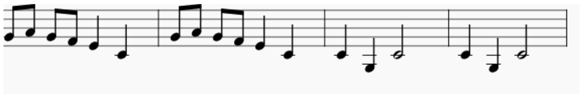
Each time is highlighted in a different colour.

- Let’s look at the music for this piece.

  Here are the first four bars:



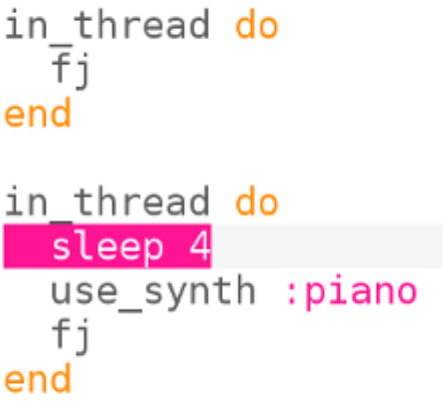
  And the final four bars:



  Run your Sonic Pi project again and follow along.

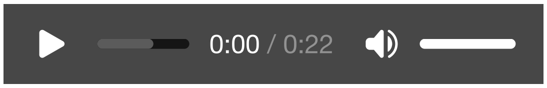
- Frere Jacques is a musical round. It’s designed to sound good when multiple versions of it start at different times. You might have been involved in singing or playing a round in music lessons at school.

  Let’s add a sleep before the piano starts playing:

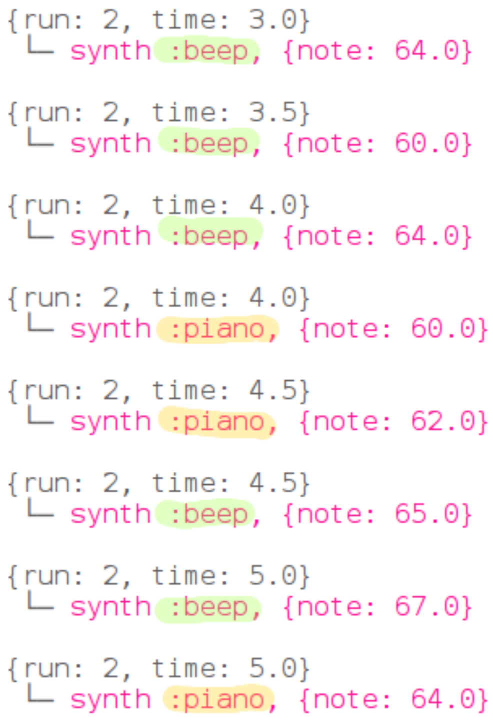


```
in_thread do
  fj
end

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

  How does it sound?



- Look at the output from Sonic Pi, can you see when the piano starts playing? And when the first instrument stops playing?



```
{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}
```

This is just an excerpt, look at your Sonic Pi output to see the whole piece.

Step 4 Challenge: More instruments

Can you add two more instruments (synths) playing Frere Jacques so that each waits another 4 beats?

Step 5 Challenge: More concurrency

What else can you do by playing multiple pieces of music at the same time using `in_thread do`?

You could find the music for another round (such as London’s Burning) and program that.

Or you could program a tune and then add rhythm in another thread.