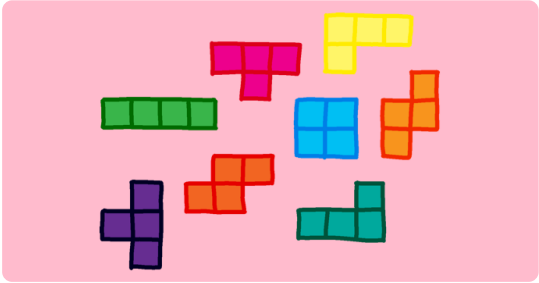
Projects

Code the Tetris Theme Tune

Program Sonic Pi to play musical notes and create the Tetris theme tune.



Step 1 Introduction

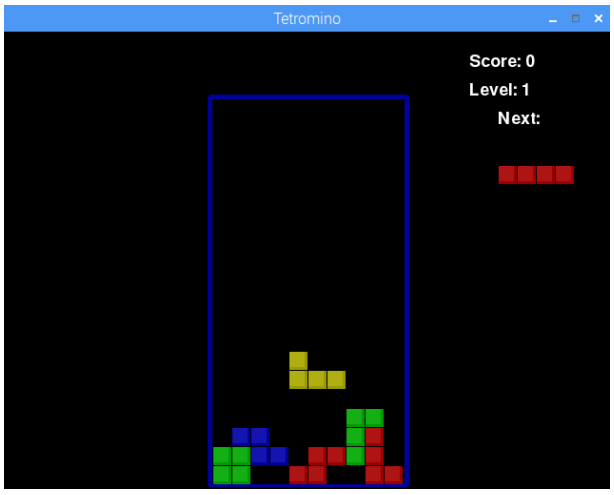
In this project you will learn how to program musical notes in Sonic Pi so that you can turn sheet music into code. You'll create your own mix of the Tetris theme tune.

Click the play button below to hear how the music will sound:

0:00 / 0:57

Tetris is a popular computer game created in 1984 by Russian coder Alexey Pajitnov. Many versions of Tetris use the music 'Korobeiniki', a Russian folk dance.

If you're using a Raspberry Pi computer then you can hear the music by playing the Tetromino game. You can find it by choosing 'Games' from the menu and then 'Python Games'. Don't play for too long! You've got coding to do.



The shapes of the Tetris pieces are called tetrominoes - the 2D shapes that can be made with 4 squares.

Additional information for club leaders

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

Club leader notes

Introduction:

In this project, children will program the Tetris Theme using musical notes and work with musical notation.

Resources

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

Club leader Resources

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

- kobeiniki.txt
- kobeiniki.mp3

Learning Objectives

- Programming music using letter names.
- Using `play_pattern_timed` in Sonic Pi.

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

- "Finish the tune" - Program more music from annotated musical notation.
- "Mix the tune" - Experiment with using different synths.
- "More of the tune" - Program more music from annotated musical notation and using `:r` for rests.
- "More music" - Find simple sheet music online and program it.

Frequently Asked Questions

Project materials

Club leader resources

- Downloadable completed Sonic Pi project (https://projects-static.raspberrypi.org/projects/tetris-theme/eb457937418d996b6e1c046bc0e03a122d4b9540/en/resources/korobeiniki.txt)
- Downloadable completed project mp3 file (https://projects-static.raspberrypi.org/projects/tetris-theme/eb457937418d996b6e1c046bc0e03a122d4b9540/en/resources/korobeiniki.mp3)

Step 2 Code the Beginning

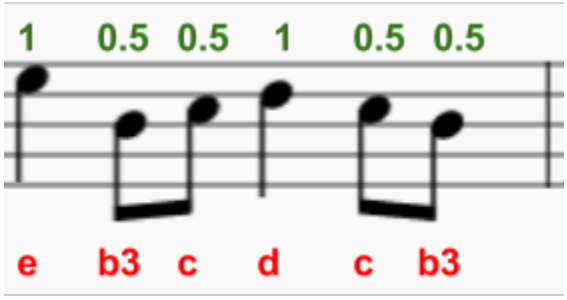
Now let's code the first part of the tune in Sonic Pi.

- First let's choose a speed and a synth for the music. The normal speed is 60 beats per minute (bpm) but that's not fast enough for this piece.

  Choose an empty Buffer in Sonic Pi and add this code:

```
use_bpm 120
use_synth :dpulse
```

- Here's the first bar of Korobeiniki:



  It shows the name of the note below in red and the length of the note above in green.

  Musical notes have letter names from A-G. To get more notes you repeat the letters to get more octaves (ranges of higher or lower notes.) Each octave starts from C.

  In Sonic Pi the default is octave 4, b3 means b from the octave below.

  Sonic Pi allows you to use letter names instead of numbers. This is useful when you are working with musical notation.

  Code the first 3 notes of Korobeiniki in Sonic Pi:

```
use_bpm 120
use_synth :dpulse

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

- That works but it takes quite a lot of typing. There's a shorter way to program longer tunes: `play_pattern`.

  `play_pattern` allows you to program multiple notes in one line.

  Replace your code to use `play_pattern` to play the first bar:

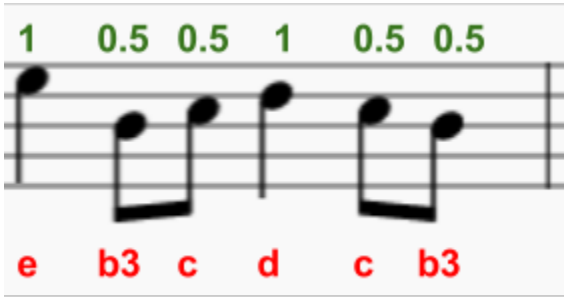
```
use_bpm 120
use_synth :dpulse

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

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

- You might have noticed that the notes aren't actually all the same length. That's okay, if you change `play_pattern` to `play_pattern_timed` then you can say how long each note lasts.

  The numbers in green show many beats each note lasts.



  (If you read music, this piece is in 4/4 time and a crotchet lasts one beat, a quaver lasts half a beat and a minim lasts two beats.)

  `play_pattern_timed` takes a list of notes and then a list of times.

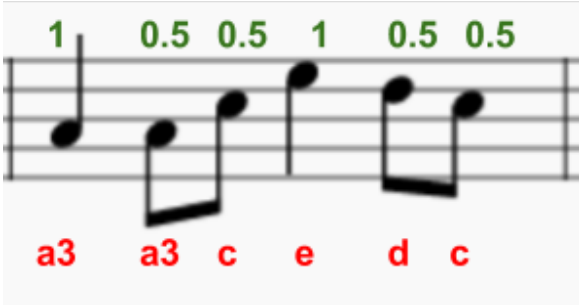
  Change your `play_pattern` code to look like this:

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

- If you don't give enough times then Sonic Pi will repeat them. This bar repeats the timing `1, 0.5, 0.5` (crotchet, quaver, quaver) so you can change your code to:

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

- Add the next bar of music, the timing is the same for this bar.



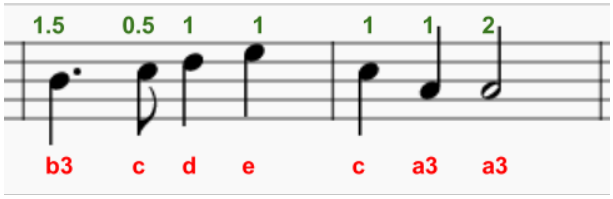

```
play_pattern_timed [:a3, :a3, :c, :e, :d, :c], [1, 0.5, 0.5]
```

0:00 / 0:07

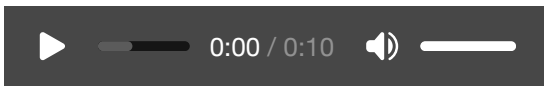
Is the tune starting to sound familiar?

Step 3 Challenge: Finish the tune

Here's the music for the next two bars.



Can you use `play_pattern_timed` to program it?



Step 4 Faster!

Russian folk songs often get faster and faster. The Tetris theme tune does this too as you get to higher levels in the game.

Let's make your tune go faster and faster.

- Your code should now look like this:

```
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]
```

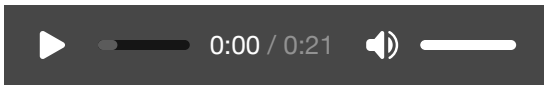
- Let's put the music into a loop first so that it plays multiple times:

```
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 can use the variable `current_bpm` to find out what the bpm is currently set to.

  Add a line to increase the bpm every time you repeat the loop:

```
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
```

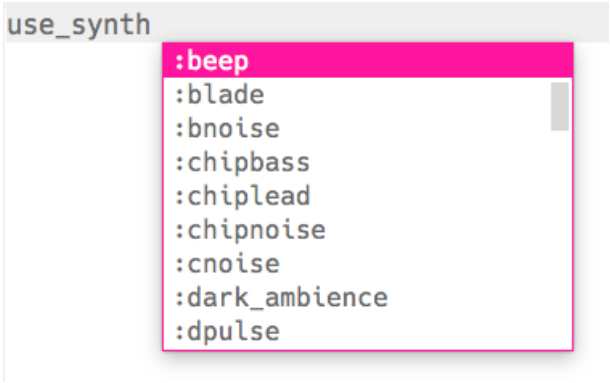


Try increasing the number of repeats to go faster and faster!

Step 5 Challenge: Mix the tune

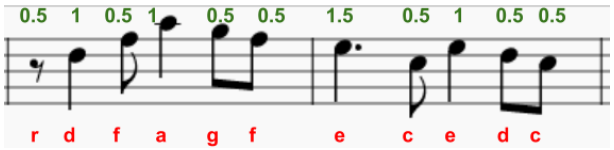
Try changing the way the tune sounds by using a different synth.

You can see a list of the synths available to you by typing `use_synth [space]` and choosing from the list that appears.

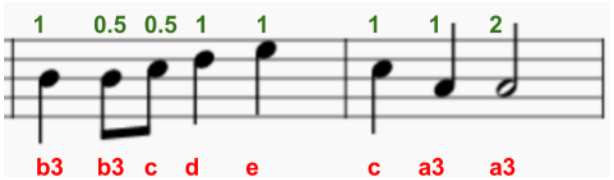


Step 6 Challenge: More of the tune

Can you program the next part of the tune:



Sonic Pi uses ':r' in lower case for a rest (a pause) and you can use it just like a note.



0:00 / 0:56

Step 7 Challenge: More music

Can you use what you've learned to program more music? You can search for sheet music online. Some music gives the letter names for notes which makes it easier.