Getting to Grips with Raspberry Pi
Sonic Pi Experimentation Module

TechResort Encounters #5138 B_SonicPi

What do I need?

- Raspberry Pi computer & power supply
- Cables and connectors to connect to monitor, mouse and keyboard
- Headphones

In the introduction to Sonic Pi you've already done you covered quite a lot.

We're now going to add further experimentation

How are we going to be using Sonic Pi?

- Again, we'll concentrate on live_loops and ".times do" loops but you can delve into the help in the Sonic Pi or into a book about Sonic Pi that we have in PDF form (ask the Elves).
- Remember that this is all an art form – there are no right answers.
- We'll point you in the direction of things that we think sound good together but there's nothing to stop you trying far more radical things…

At the end you'll output music files to a standard format which you can play on general media players.

Turn over to get going!

Getting Started

- Set up your Raspberry Pi as you did before – using your Raspbian image on the SD card.

- Check back to the introductory script, or ask an elf if you can't remember how to do it.

Important

Remember to power up the Pi last!

- Now start up Sonic Pi

- You should see some of the code that you wrote in the last session. Why not play a bit and familiarise yourself with it all again?

Building a new piece

We've prepared a bass riff. The bass line in a piece of music is often the first thing that gets developed.

Open the SonicEx03.rb file into a buffer with nothing else in it

At the moment – this piece of code won't work. See if you can fix it.

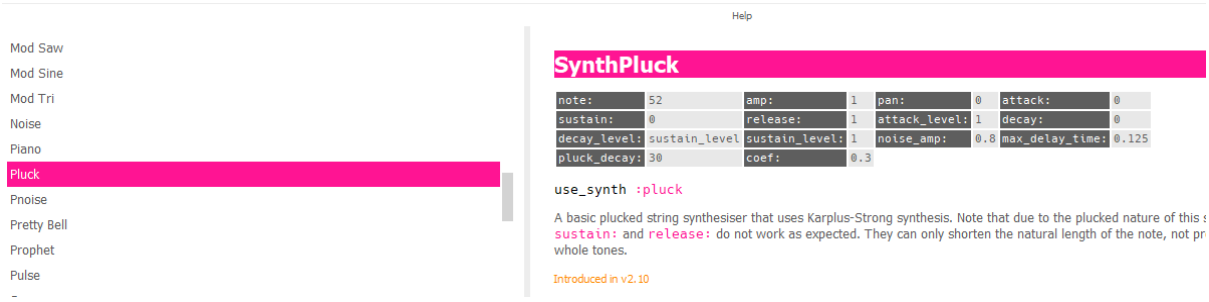
(hint: how long do all the beats play for? Where is it defined?)

Now play it.

The only line of code that might be unfamiliar is the:

```
use_synth_defaults
```

command and its parameters. Once the riff is working try tinkering with the values of the options that are listed after the command.



Notes:

- Go to the help-synths section at the bottom of the screen. Select "pluck" and you should see information about the parameters for SynthPluck and advice on how to use them. Scroll down below the table of information we've shown above for more information about each different option specifically for this particular synthesiser voice
- Attack, decay and release are all parameters which shape how the note sounds – they all need to be zero or higher.
- Res is a number which indicates how much resonance there is on the sound.

Other experimentation

- Use different synths and add new synth defaults to see how they sound.

When you're happy with the riff make sure you save your buffer to your Sonic Pi folder.

Treble Riff

We've prepared another riff for you that goes with this one.

Load it into an empty buffer from SonicEx04.rb

Copy and paste the code into your working copy (where the bass riff is).

Check it works

Save your code for your working buffer.

More experimentation

- You'll see this new piece of code uses a different synth voice. Does it work, do you think? Try some others.
- You can also tinker with the `synth_default` values.

Percussion

Have a go at adding some percussion. You should have noticed we've put some comments at the beginning of each loop with information about it – including how many beats (or basebeats) long it is. They're always a multiple or a factor of four so that they fit with the underlying beat of the music

- Create a new live_loop and call it a sensible name (check other loops if you need a reminder for how the code works).
- It's worth adding the live_loop end statement as soon as you create a new live_loop just to make sure you don't forget it (properly indented of course)
- Now add a ".times do" loop (and the end statement)
- Trigger a sample of your choice with a

  ```
  sample :xxxxxx
  ```

  statement (where xxxxxx is the name of the sample)
- Then add a sleep statement for the duration you'd like (don't forget to use the basebeat variable)
- You can do several ".times do" loops one after the other so your percussion becomes more interesting and complex

Your code will probably end up looking something like this

```
# Welcome to Sonic Pi v2.11.1
#Now the underlying drums - 16 basebeats long
live_loop :bassdrum do
  6.times do
    sample :drum_heavy_kick
    sleep basebeat * 2
  end
  8.times do
    sample :drum_heavy_kick
    sleep basebeat * 0.5
  end
end
```

- Note that in my example `:bassdrum` is the name of the loop
- I'm using the sample "`:drum_heavy_kick`"
- The total number of basebeats in the two loops combined is 16 because it's

  6 x 2 x basebeat = 12
  8 x 0.5 x basebeat = 4

More ornamentation

- You can add more and more `live_loops` to add extra interest to your piece
- You can put long sleep commands at the beginnings of live_loops to make sure some loops come into the music very late.
- If you look at our file SonicEx06.py (by loading it into a fresh buffer) you'll see lots of different bits and pieces we've added.
- Feel free to tinker with it and experiment but remember to save each buffer file to your SonicPi directory to ensure you don't accidentally lose your work

Chords

So far we've done some loops with notes and loops with samples. We've also covered two sorts of loops which can be used to make the music sound more intricate.

We're now going to introduce chords which are used to fill out the sound of the music, making it richer.

We're going to start with a new example

- Open SonicEx07.rb into an empty buffer
- Play it to make sure it works.

After defining the basebeat variable which we'll use to determine the speed of the music (like before), I've defined some chords.

Chords are groups of (usually) three or more notes which will get played together. In music theory there are lots of conventions about chords but the truth is, you can combine any notes you like – it's just that if you're not careful, the notes don't work together very well.

I've defined four conventional chords near the beginning of my code.

Once I've defined them, I can use them like notes with a "play" command – but notice that you don't prefix the name of the chord with a colon like you would with

```
play :B3
```

You use the name of the chord we set up earlier in brackets

```
play (chord1)
```

Chords work really well when instead of being a short sound that dies away quickly (like some of the bass lines we've been using), they're played in a voice which resonates a bit, lasting longer.

Challenges

- Add a good synth voice and tinker with the defaults for it in the chords loop. Try and get a nice rich sound.
- Listen to the music for long enough that the final live_loop (called melody1) plays once.

  It was supposed to play for the first time after the chord sequence had played through twice. What's going wrong? Can you fix it.
- See you if you can improve the drums – at the moment they're OK, but the rhythm's a bit dull
- Can you find a good synth voice (and default values, perhaps) for the melody?
- Add other melody lines – perhaps timing them to fit around melody1
- Keep experimenting as much as you like

Don't forget to save your work (each buffer separately)

Can I play my music to other people?

Yep! We can get Sonic Pi to output a standard music-type file that can be played by computers and other music applications

You've been using the buttons at the top of the screen for running, stopping, saving and loading your code. We're going to use the "Rec" (or record) button.

Press the Rec button, then shortly after press the "Run" button.

If you're using a loop let your loop run for as many times as you'd like to incorporate any live loops that only crop up from time to time.

Now press stop to stop your music from playing and then press Rec again to stop recording (this way you should get a short gap at the end of the music file.

You should now see a dialog box in the middle of the screen called "Save Recording" – as usual for saving files, navigate to the folder you want to put it in, and name your file something meaningful.

Notice that the file type is "Wavefile" or .wav

This is the only file type that SonicPi will save audio files as but most audio file players (like Windows media player, Groove Music and iTunes) will happily play these files. Note also, that .wav files are pretty big.

Once you've saved your file you can copy it onto a USB stick if you have one but you'll be able to take home the USB stick you've been using at the end of the 6-week programme.