Making Music through Computer Science Practices

Jared O'Leary
BootUp PD

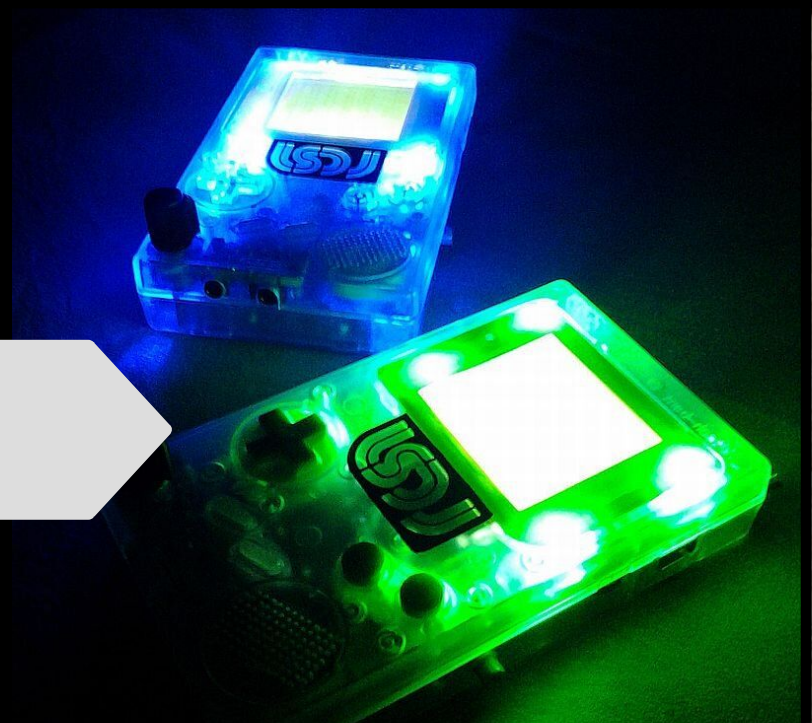
How to reach the resources

- Direct link is in the chat
- www.JaredOLeary.com
  - Presentations
    - Making Music through Computer Science Practices

Computer Science

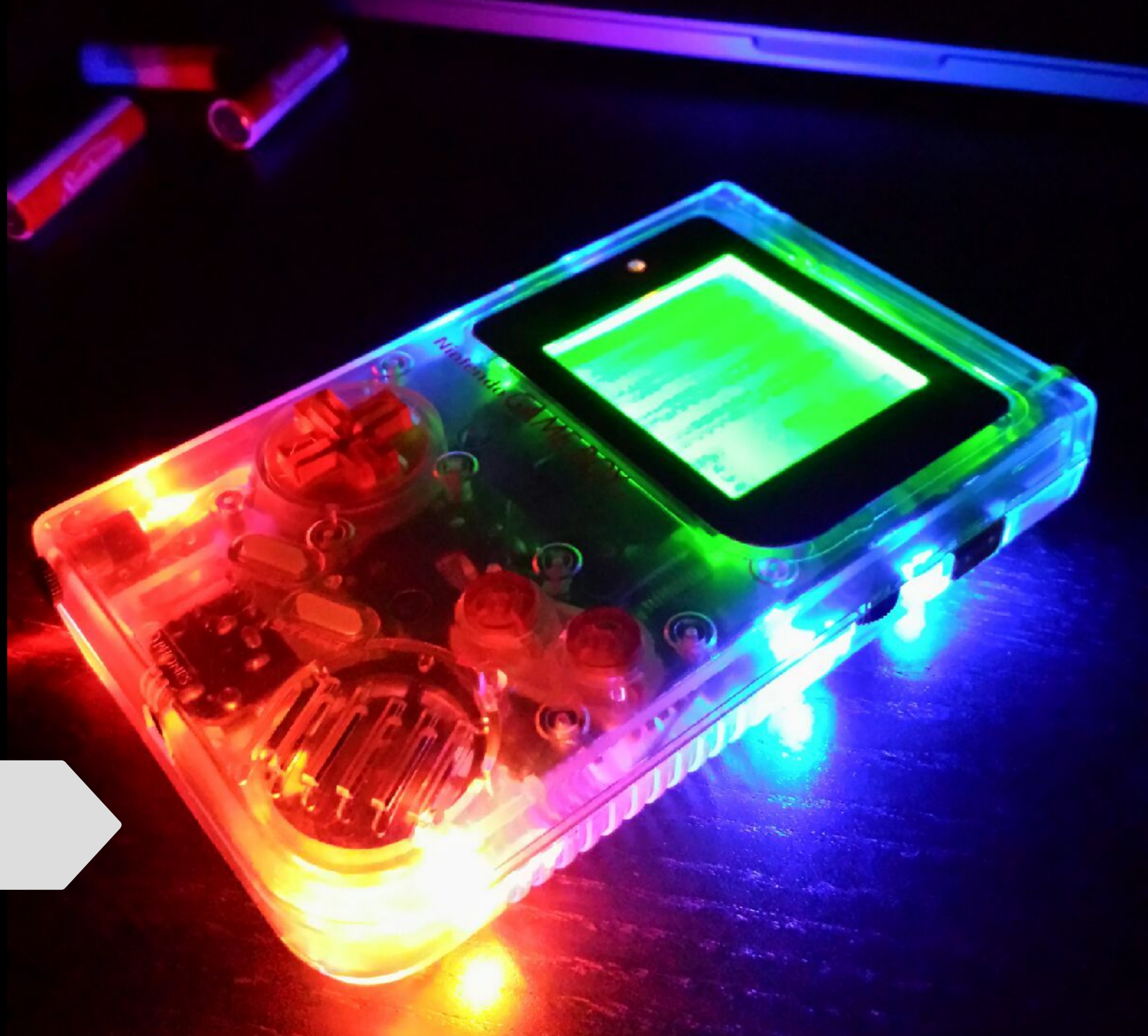
Performance practices

Maker practices



Aesthetic mods

LEDs



Functionality mods

Backlighting

Functionality mods

Prosound

Functionality mods

Prosound

Functionality mods

Clocking

Functionality mods

Circuit-bending

Functionality mods

Circuit-bending

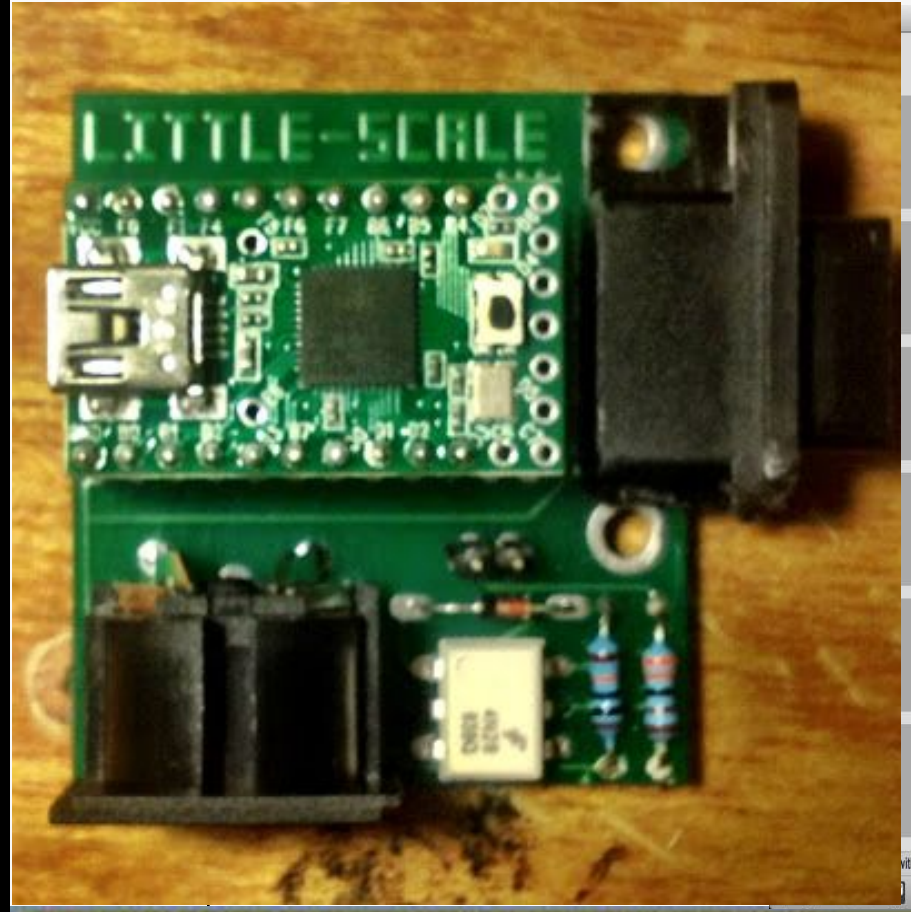
Functionality mods

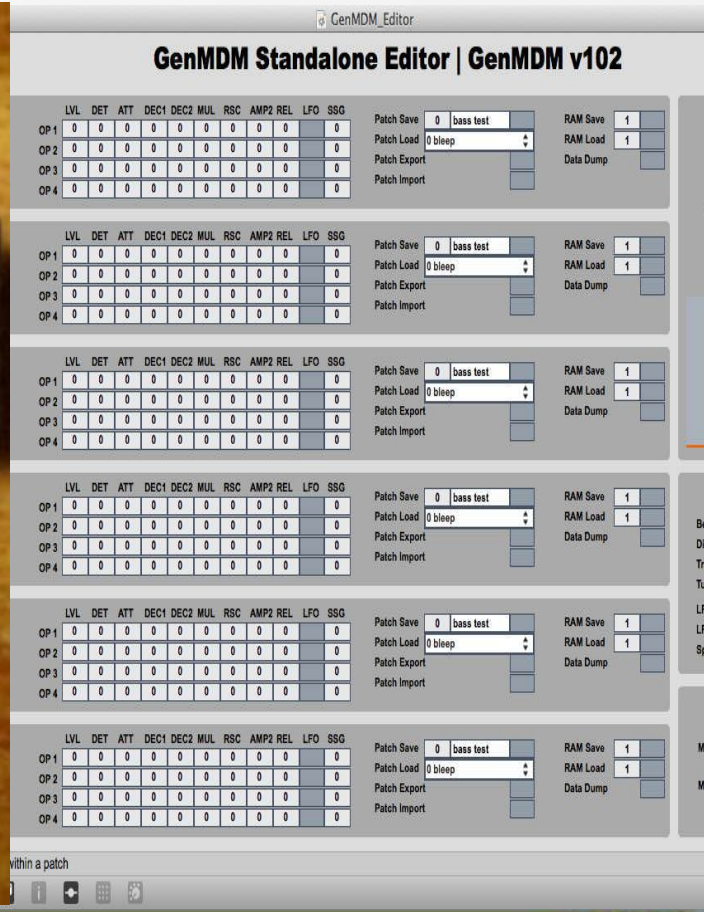
Other mods

Functionality mods

Other mods

Manufacturing or building new devices





Coding practices

Here's how I'd do it, starting with the version at https://github.com/trash80/Arduinoboy;

In the file Mode.ino edit the function switchMode() as follows;

```
void switchMode()
{
  switch(memory[MEM_MODE])
  {
    case 0:
      modeLSDJSlaveSyncSetup();
      break;
    case 1:
      modeMidiGbSetup();
      break;
  }
}
```

And in the file Arduinoboy.ino, change the line;

```
#define NUMBER_OF_MODES 7    //Right now there are 7 modes, Might be more in the fut
```

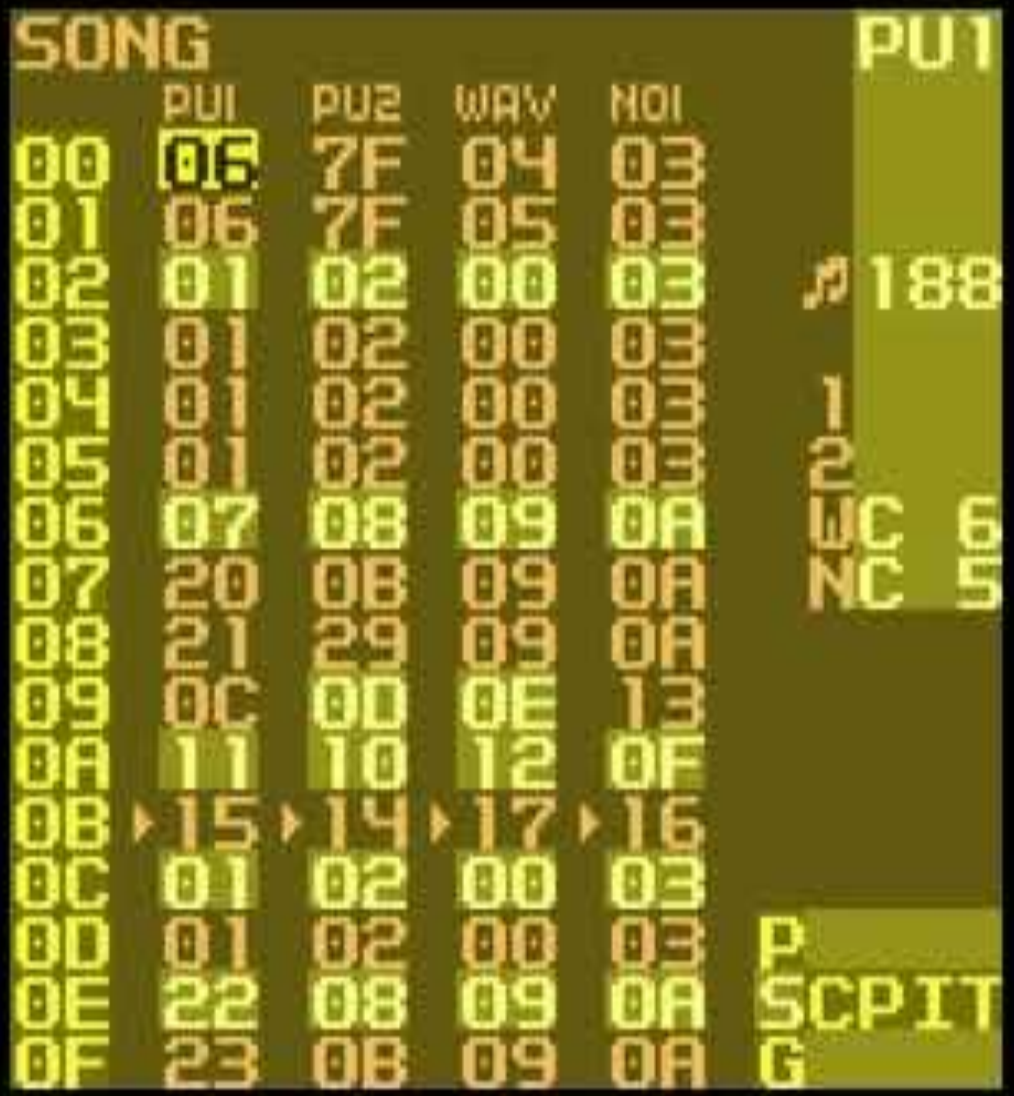
to

```
#define NUMBER_OF_MODES 2
```

That should pretty much do what you want.

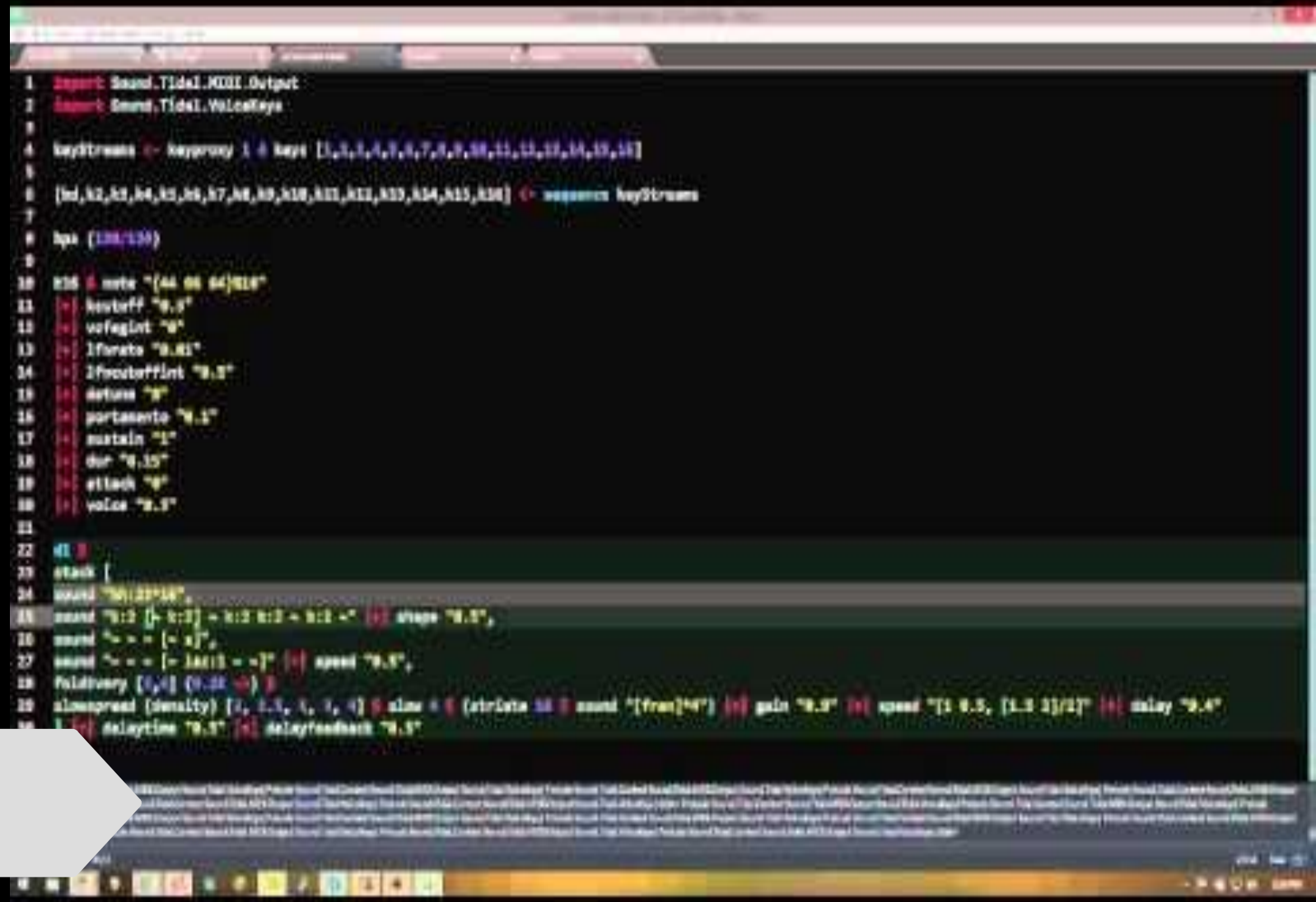
Software development

Software development



Creating music with code

Live coding



Hot Cross Buns (with Sonic Pi)

Setting our tempo

1. use_bpm 144

Adding our notes

1. use_bpm 144
2.
3. play :e
4. play :d
5. play :c

Separating our notes

1. use_bpm 144
2.
3. play :e
4. sleep 2
5. play :d
6. sleep 2
7. play :c
8. sleep 4

Defining a function

```
1.  use_bpm 144
2.
3.  define :buns do
4.    play :e
5.    sleep 2
6.    play :d
7.    sleep 2
8.    play :c
9.    sleep 4
10. end
```

Congratulations, you recreated Cage's *4' 33"*!

Calling our function

```
3.  define :buns do
4.    play :e
5.    sleep 2
6.    play :d
7.    sleep 2
8.    play :c
9.    sleep 4
10. end
11.
12. buns()
13. buns()
```

Starting our next phrase

12. buns()
13. buns()
14.
15. play :c
16. sleep 1

Using repeats

```
12.  buns()
13.  buns()
14.  4.times do
15.    play :c
16.    sleep 1
17.  end
```

Using repeats

```
12.  buns()
13.  buns()
14.  4.times do
15.    play :c
16.    sleep 1
17.  end
18.  4.times do
19.    play :d
20.    sleep 1
21.  end
```

Completing our song

```
12.  buns()
13.  buns()
14.  4.times do
15.    play :c
16.    sleep 1
17.  end
18.  4.times do
19.    play :d
20.    sleep 1
21.  end
22.  buns()
```

Changing our synth

```
1.  use_bpm 144
2.  use_synth :tri
3.
4.  define :buns do
5.    play :e
6.    sleep 2
7.    play :d
8.    sleep 2
9.    play :c
10.   sleep 4
11. end
```

Shaping our notes

```
1.  use_bpm 144
2.  use_synth :tri
3.
4.  define :buns do
5.    play :e, release: 2
6.    sleep 2
7.    play :d, release: 2
8.    sleep 2
9.    play :c, release: 4
10.   sleep 4
11. end
```

Adding effects

```
13.
14.     with_fx :echo do
15.         buns()
16.         buns()
    . . . . . .
24.         buns()
25.     end
```

In a different buffer

1. use_bpm 144

Creating our loop

1. use_bpm 144
2.
3. live_loop :perc do
4. end

Metal

```
1.  use_bpm 144
2.
3.  live_loop :perc do
4.    sample :bd_haus
5.    sleep 0.25
6.  end
```

EDM

```
1.  use_bpm 144
2.
3.  live_loop :perc do
4.    sample :bd_haus if (spread 1, 4).tick
5.    sleep 0.25
6.  end
```

Adding in another rhythm

```
1.  use_bpm 144
2.
3.  live_loop :perc do
4.    sample :bd_haus if (spread 1, 4).tick
5.    sample :elec_bong if (spread 3, 8).look
6.    sleep 0.25
7.  end
```

…and another

```
1.  use_bpm 144
2.
3.  live_loop :perc do
4.    sample :bd_haus if (spread 1, 4).tick
5.    sample :elec_bong if (spread 3, 8).look
6.    sample :perc_snap if (spread 3, 4).look
7.    sleep 0.25
8.  end
```

Adjusting our amplitude

```
1.  use_bpm 144
2.
3.  live_loop :perc do
4.    sample :bd_haus if (spread 1, 4).tick
5.    sample :elec_bong if (spread 3, 8).look
6.    sample :perc_snap, amp: 0.3 if (spread 3, 4).look
7.    sleep 0.25
8.  end
```

Back in our original buffer

```
12.
13.     define :song do
14.       with_fx :echo do
15.         buns()
16.         buns()
   . . . . . .
25.         buns()
26.       end
27.     end
```

Press Run for Cage's encore

(there is a purpose for this)

Hip cross buns

```
1.  use_bpm 144
2.
3.  live_loop :perc do
4.    sample :bd_haus if (spread 1, 4).tick
5.    sample :elec_bong if (spread 3, 8).look
6.    sample :perc_snap, amp: 0.3 if (spread 3, 4).look
7.    sleep 0.25
8.  end
9.
10. song()
```

Exploring Sonic Pi

- Sonic Pi's built-in help
  - Tutorials
  - Examples
  - Synths
  - Fx
  - Samples
  - Lang(uage)
- www.JaredOLeary.com/sonic-pi

Let's Share What We Created!