Still, Inside

for live coder

Cameron McArthur

March 2021

Still, Inside

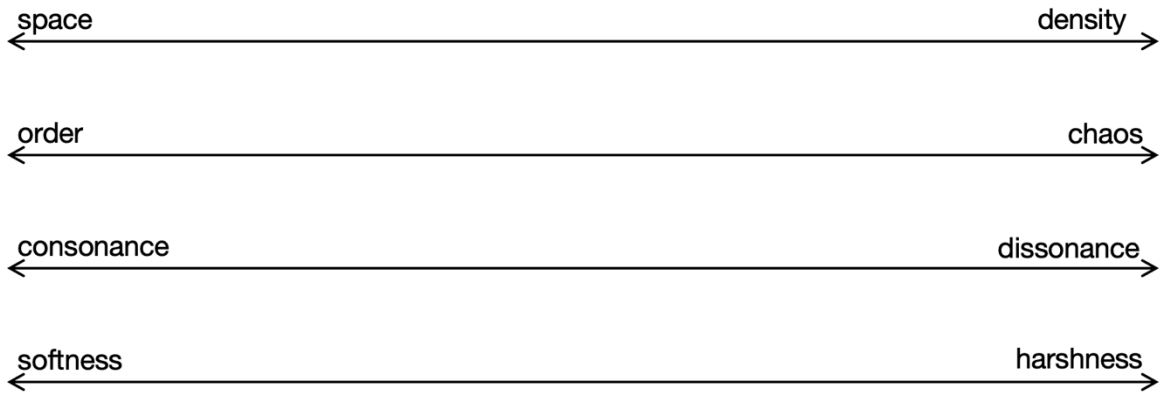
Composer's Note

Still, Inside was created in the early months of 2021. It is a reflection on my experience of the first Covid–19 lockdowns, and the impact that isolation had upon my relationship with the everyday objects around me. Lockdowns, social distancing, and the move to home–working rapidly and significantly altered the landscape of music–making in general. Once they were out of reach, the hard and unfriendly buildings outside my window suddenly seemed a lot more inviting. My perception of time seemed to warp and twist, as I studied those buildings from new angles. Then, as restrictions eased and the world opened up again, I found myself seeing and hearing new details in the buildings, objects, people, and places that I had always taken for granted. In *Still, Inside*, I tried to capture some of that experience, sampling, transforming, and reconstructing familiar instrumental sounds to reveal new qualities and overlooked features through temporal and conceptual engagement with momentum.

Performance Instructions

The material for *Still, Inside* is intended for use in the music creation software tool Sonic Pi. Blocks of code are separated into six organisational buffers. These blocks of code are designed to facilitate various kinds of randomisation, and multiple ways of controlling the sounds. One block of code might command Sonic Pi to choose a particular pitch, and to play this pitch for a certain duration, or to pass it through FX such as reverb, delay, ring modulation, to alter the sonority. Another might involve only unpitched samples and randomise their playback rate (speed). These values (and many others) can all be adjusted at will, during performance – their specific temporal placement and continuation is at the performer's discretion.

The basic principle for this work is simple – the performer should manipulate the coded material, imagining that they are moving across the axes shown below. At first, they should move from left to right, gradually transforming space, order, consonance, and softness into density, chaos, dissonance, and harshness, directing the music towards its climax. Then, once they feel that process has run its course, they should make similarly gradual movement back towards the starting point, reducing the intensity and slowing the rate of change, until the work is ready to be brought to its close.



The pseudo-randomisation built into the coded material ensures that, though the coder will always know what the approximate outcome of a command will be, they cannot make specific decisions regarding pitch content, or be overly rhythmically, timbrally, or temporally precise. This is intentional, and the local imprecision is a desirable trait. In this way, any two given performances should communicate a broadly similar sense of momentum, but each performance should be unique, incorporating a sizeable amount of local, targeted randomness into the overall shape.

The Role of the Score

The notated score included in this document is not intended for use in performance. The performer should familiarise themselves with the sampled material, and then use the axes as guides for their decision-making. Instead, the score is a documentary reflection of the original version of this piece, which was devised in collaboration with Lizzie Knatt (recorders) and Federico Pendenza (guitar). It shows the approximate moments at which the coded blocks were introduced in that audiovisual version of the work, outlining the basic shape and flow. Though this might serve as a useful point of reference for live coders looking to perform the piece, I encourage performers to make their own intuitive decisions regarding the placement and combination of sounds, focused on creating momentum through movement across the axes.

Buffers

The code used in the original performance is included below for reference. **Note:** Performers must update the sample file paths to match the file locations on their own devices or storage systems.

Buffer 0

```
in_thread do
  with_fx :level, amp: 0  do
    live_loop :pedal_a do
      sample "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/whistle tone a.wav",
        rate: choose([ -0.5, -1, 0.5, 1]), amp: rrand(0.8, 1.5), pan: rrand(-1, 1)
      sleep rrand(6, 8.9)
    end
  end
end

sleep 10

in_thread do
  with_fx :level, amp: 0.00, amp_slide: 50 do |a|
    live_loop :whistle_hex_1 do
```

```
    sample choose([ "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/whistle tone a flat.wav",
                    "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/whistle tone a.wav",
                    "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/whistle tone c sharp.wav",
                    "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/whistle tone d.wav",
                    "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/whistle tone e.wav",
                    "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/whistle tone g.wav"
                     ]), rate: choose([ -0.5, -1, 0.5, 1]), amp: rrand(0.8, 1.5), pan: rrand(-1, 1)
    sleep 3
  end
  control a, amp: 0.2
  sleep 3.5
  control a, amp: 0.5
  sleep 3.5
  control a, amp: 0.8
  sleep 3.5
  control a, amp: 1
 end
end
```

Buffer 1

```
#run after approx. 70 secs
with_fx :level, amp: 0 do

 with_fx :gverb, mix: 0.3 do
  sample "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/phrase 1.wav", rate: -0.5, pan: -0.5
  sleep rrand(8, 16)
```

```
    sample "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/phrase 4.wav", rate: -0.5, pan: 0.4
    sleep rrand(8, 16)

    sample "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/phrase 2.wav", rate: -0.5, pan: 0.2
    sleep rrand(8, 16)

    sample "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/phrase 3.wav", rate: -0.75, pan: -0.35
    sleep rrand(8, 16)

  end
end
```

Buffer 2

```
#run when last phrase in buffer 1 is heard
live_loop :destroyed_motif do
  with_fx :level, amp: 1 do
    with_fx :gverb do
      with_fx :pitch_shift, pitch: rrand(-12, 12) do
        sample choose(["/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Harmonics.wav",
        "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/motif 1.1.wav",]),
          rate: rrand(-1, 1), pan: rrand(-1, 0)
        sleep rrand(3, 15)
      end

    end
  end
end
```

Buffer 3

```
#run like dripping rain
```

```
live_loop :dyads do
  with_fx :level, amp: 0.5 do
    with_fx :autotuner do |a|
      with_fx :bitcrusher, sample_rate: 44100, sample_rate_slide: 3, bits: 16, bits_slide: 0.3 do
|b|
        with_fx :whammy, transpose: rrand(0, 12), transpose_slide: 2 do |w|
          with_fx :krush do
            sample choose(["/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Dyads/D_1.wav",
                           "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Dyads/D_2.wav",
                           "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Dyads/D_3.wav",
                           "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Dyads/D_4.wav",
                           "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Dyads/D_5.wav",
                           "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Dyads/D_6.wav",
                           "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Dyads/D_7.wav",
                           "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Dyads/D_8.wav",
                           "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Dyads/D_9.wav",
                           "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Dyads/D_10.wav",
                           "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Dyads/D_11.wav",
                           "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Dyads/D_12.wav",
                           "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Dyads/D_13.wav",
                           "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Dyads/D_14.wav",
```

```
                    "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Dyads/D_15.wav",
          "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Dyads/D_16.wav"]),
            amp: 1, pan: rrand(-1, 1), rate: choose([ -0.5, -1, 0.5, 1])
          control b, bits: 8, sample_rate: 3000
          control w, transpose: rrand(-24, 24)
          control a, note: choose([:e2, :e3, :a2, :a3, :a4, :gs3, :gs4, :c4, :c5, :g2])
          sleep rrand(1.2, 2.9)
        end
      end
    end
  end
end
```

Buffer 4

```
#time is stretching
live_loop :guitar_unpitched_sounds do
  with_fx :krush do
    with_fx :gverb, mix: 0.3 do
      sample choose(["/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Unpitched/Big tap.wav",
                    "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Unpitched/Small Taps.wav",
                    "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Unpitched/String Bristle.wav",
                    "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Guitar Samples/Final/Unpitched/String Scratch.wav"

                    ]),rate: rrand(-0.5, 2), pan: rrand(-0.5, 0.5), amp: 1
      sleep rrand(3, 16)
      with_fx :normaliser, level: 0.7 do
      end
    end
```

```
  end
end
```

Buffer 5

```
#time, distorted
live_loop :multiphonics do
  with_fx :krush do
    with_fx :gverb do
      sample choose(["/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/multiphonic 6-7.wav",
                     "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/multiphonic 1.wav",
                     "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/multiphonic 2.wav",
                     "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/multiphonic 3.wav",
                     "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/multiphonic 4.wav",
                     "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/multiphonic 5.wav"
                     ]), rate: rrand(-0.5, 0.5), pan: rrand(-0.8, 0.8), amp: 1
      sleep rrand(1, 2)

    end
  end
end
```

Buffer 6

```
#fading
use_random_seed 12
in_thread do
  with_fx :level, amp: 1  do
    with_fx :krush do
      with_fx :pitch_shift, pitch: -36 do
        live_loop :pedal_a do
```

```
        sample "/Users/cameron/Documents/Projects/Ongoing/Arc Project/Digital
Ensembles/Recorder Samples/multiphonic 1.wav",

        rate: choose([ -0.5, -1, 0.5, 1]), amp: rrand(0.8, 1.5), pan: rrand(-1, 1)

        sleep rrand(3, 5.9)
      end
    end
  end
 end
end
```

Still, Inside

Documentary score

Cameron McArthur