

# Generating Melodies with Series and Trignometrics

---

## Abstract

We encounter a lot of sounds in daily life, and the sound made by musicians is especially interesting.

The sequence of Do. Re. Mi. seems that it has some kinds of magic. I'm always amazed by the elegant of music

and eager to reveal some secret of it. These *Mathematica* snippets do a trivial job to peek the pattern of melodies that we enjoy, and how to generate them.

---

## Import some functions

```
Import["E:\\MyWorkShop\\midi\\notebooks\\midilib.nb"];
```

---

## First try

The first idea that came into my mind is to generate melody sequence by accumulation. For example, given a sequence  $\{x_1, x_2, \dots, x_n\}$  compute finite series of it:  $\{s_1, s_2, \dots, s_n\}$ , and by varying  $x_1$ , we can tilt the series a little( move up and down).

Then introduce a new term  $y$ , and put it in front of  $s_1$ , we get a new sequence. Next compute the series of this new sequence we get

$\{z_1, z_2, \dots, z_n\}$  ( $z_1 = y, z_2 = y + x_1 \dots$ ) and by varying  $x_1, y$  we actually can change the shape of the whole melody.

The following codes demonstrate this idea:

```
Clear[musictemplate]
```

```
In[211]:= polydemo = {1, 2, -5, 4, -3, 6, 1};
```

```
In[232]:= GenSeq[n_] := RandomInteger[{-2, 2}, n];
```

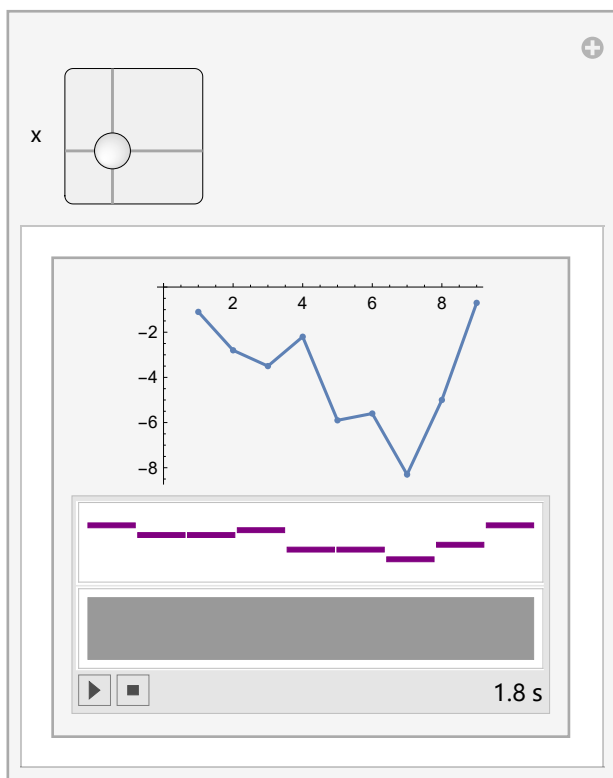
```
AccOrder2[x1_, x2_, li_] := Accumulate[{x1}~Join~Accumulate[{x2}~Join~li]];
```

```

In[235]:= Manipulate[
  Module[{seq = AccOrder2[x[[2]], x[[1]], polydemo]},
    Panel@Grid[{
      {ListPlot[
        seq,
        PlotRange → Full, Joined → True, Mesh → All
      ]},
      {Sound[SoundNote[Round[#], 0.2] & /@ seq]}
    ]},
  ],
  Control[{x, {-4, -4}, {4, 4}, 0.1}]
]

```

Out[235]=



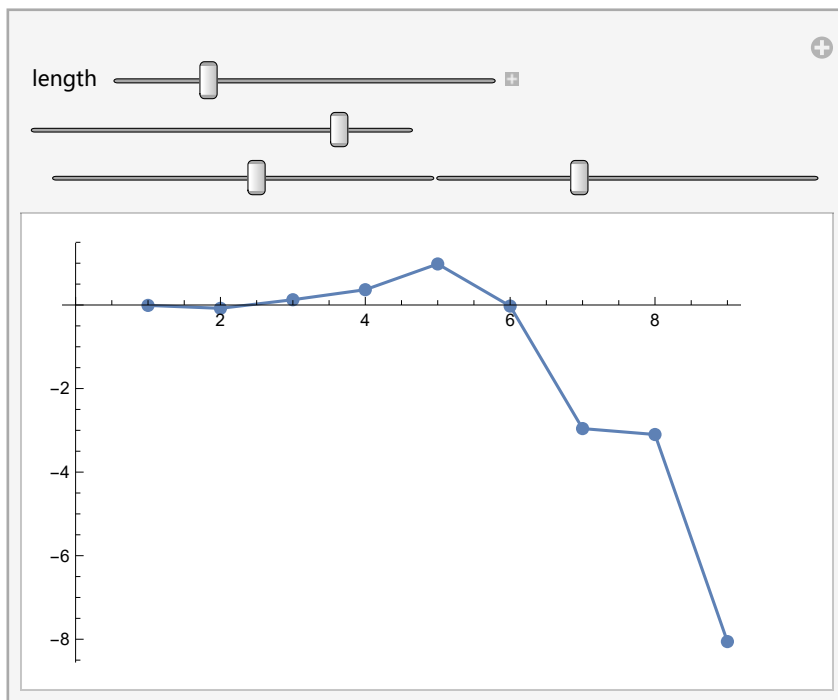
It do have some shape, but the melody isn't so fancy.

The following codes accumulates the sequence further, with higher order and introducing more control terms, but the melody is even worse!

```

Manipulate[
  data = Fold[
    Accumulate[Insert[#1, #2, RandomInteger[{1, Length[#1]}]]] &,
    {-.01, -.05, .1, -2, -0.3, -.1}, x[[1 ;; length]]
  ]; ListPlot[
    data,
    PlotRange → Full, Joined → True, Mesh → All
  ],
  {{length, 5}, 1, 10, 1},
  {{x, Table[0, {i, 1, 30}]}, ControlType → None},
  Dynamic[Row@Outer[Slider[Dynamic[x[[#1]]], {-0.5, 0.5}] &, Range[length]]]
]

```



## Generating melody with trigonometrics

Later I found that what makes a melody fascinating actually is the periodicity. Further, only periodicity isn't so cool.

Periodicity with variation is the key.

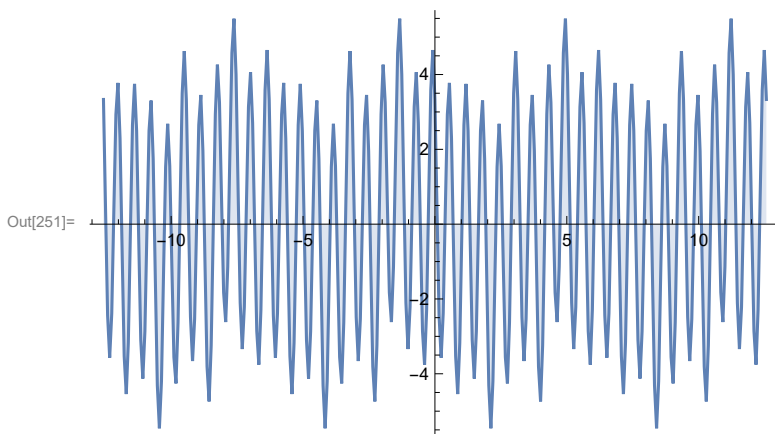
With this thought, a simple step is to create some sinusoid function and add them up, then take samples by some constant step.

So we'll get a sequence, play it and manipulate the sinusoid functions to obtain a nice melody.

The following codes demonstrate this idea.

```
In[247]:= Clear[li, l2, f1]
li =
  {Cos[x] - 2 Sin[x], 8 Cos[10 x] - 9 Sin[10 x], -Cos[3 x] + 0.6 Sin[3 x], 2 Cos[4 x]};
l2 =  $\frac{1}{3}$  Plus@@li
f1[t_] := -0.4;
DiscretePlot[l2, {x, -4 Pi, 4 Pi,  $\frac{1}{40}$  Pi}]
```

```
Out[249]=  $\frac{1}{3} (\text{Cos}[x] - \text{Cos}[3 x] + 2 \text{Cos}[4 x] + 8 \text{Cos}[10 x] - 2 \text{Sin}[x] + 0.6 \text{Sin}[3 x] - 9 \text{Sin}[10 x])$ 
```



```
Sound@musictemplate[
  List /@ (CMajor /@ Round@Array[10 + l2 /. x -> (# *  $\frac{1}{10}$  Pi) &, 100]), 0.2]
```



```
musictemplate[{{1}, {3}, {5}}, 0.2]
{SoundNote[1, {0.2, 0.44}, Piano],
 SoundNote[3, {0.4, 0.64}, Piano], SoundNote[5, {0.6, 0.84}, Piano]}

Export["E:\\MyWorkShop\\midi\\demo.mid", Sound@musictemplate[
  List /@ (CMajor /@ Round@Array[12 /. x -> (# *  $\frac{1}{10}$  Pi) &, 100]), 0.15]]

E:\\MyWorkShop\\midi\\demo.mid
```

Then represent trigonometrics in complex exp. form:

```
In[252]:= randomsinusoid[len_] := Table[RandomComplex[{-3 - 3 I, 3 + 3 I}], {i, 1, len}]
synthesize[ingre_] :=
  Plus@@MapIndexed[#1 * E2 Pi i (#2[[1]]-1) x + Conjugate[#1] * E2 Pi i (1-#2[[1]]) x &, ingre]
```

## More Examples

**randomsinusoid[8]**

```
{-2.49012 + 0.575478 i, -0.324604 + 2.38509 i,  
-0.777139 + 2.89526 i, 2.1392 - 1.26265 i, 0.0735779 + 1.1113 i,  
-1.87081 + 2.96822 i, -2.04614 - 2.51585 i, 2.43442 - 0.433813 i}
```

**synthesize[randomsinusoid[8]]**

```
(3.19648 + 0. i) + (2.5001 - 0.286782 i) e^{-2 i \pi x} + (2.5001 + 0.286782 i) e^{2 i \pi x} +  
(0.459633 + 0.176403 i) e^{-4 i \pi x} + (0.459633 - 0.176403 i) e^{4 i \pi x} -  
(0.0167887 + 2.75794 i) e^{-6 i \pi x} - (0.0167887 - 2.75794 i) e^{6 i \pi x} +  
(2.65711 - 2.2142 i) e^{-8 i \pi x} + (2.65711 + 2.2142 i) e^{8 i \pi x} -  
(2.05273 + 0.453504 i) e^{-10 i \pi x} - (2.05273 - 0.453504 i) e^{10 i \pi x} -  
(0.685545 + 0.878823 i) e^{-12 i \pi x} - (0.685545 - 0.878823 i) e^{12 i \pi x} -  
(0.56413 - 2.83456 i) e^{-14 i \pi x} - (0.56413 + 2.83456 i) e^{14 i \pi x}
```

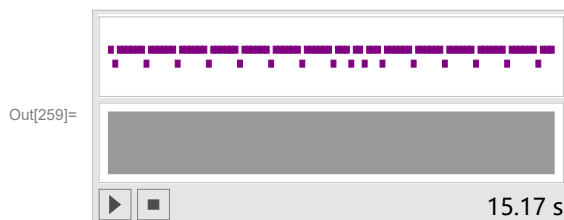
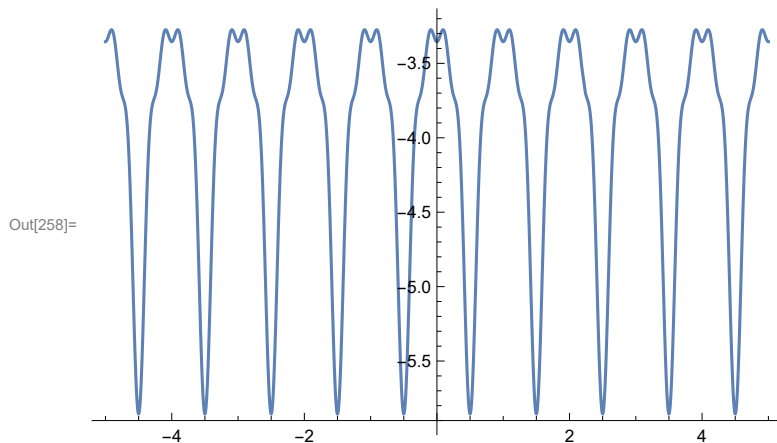
In[257]:= **tone = synthesize**  $\left\{-2, \frac{1}{2}, -\frac{2}{9}, \frac{1}{8}, \frac{-2}{25}\right\}$

**Plot[tone, {x, -5, 5}]**

**Sound@musictemplate**  $\left[$

**List /@**  $\left(\text{CMajor} / @ \text{Round} @ \left(\frac{1}{10} * \text{Array} \left[\text{tone} /. x \rightarrow \left(\# * \frac{2}{1} \text{Pi}\right) \&, 100\right]\right), 0.15\right]$

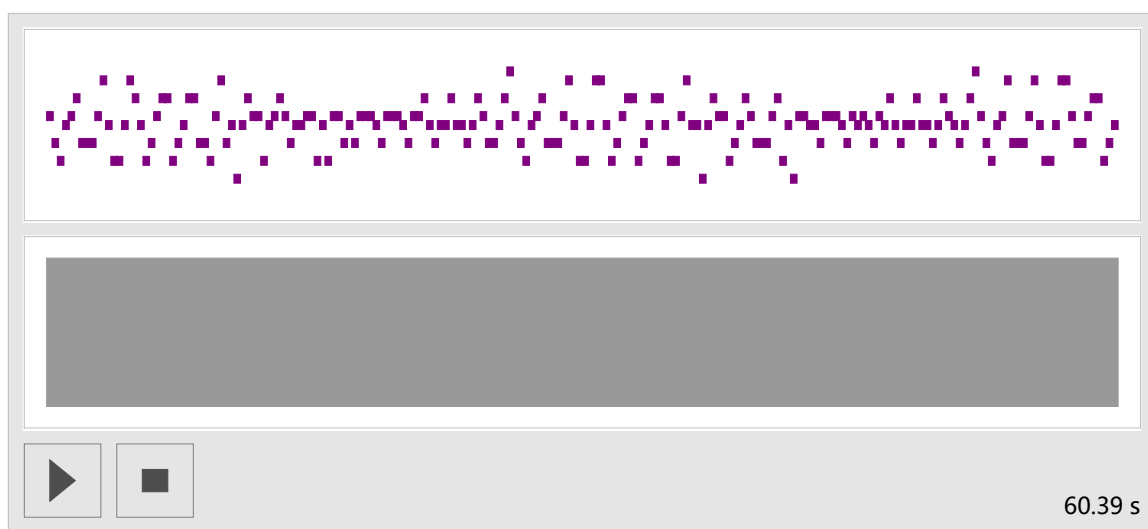
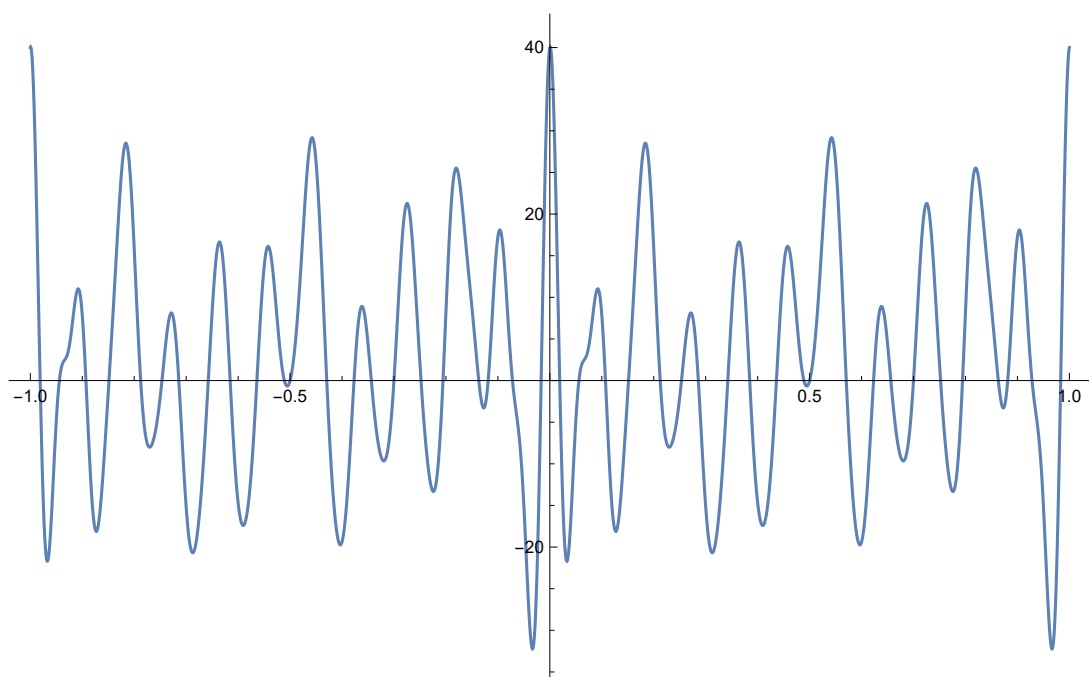
Out[257]=  $-4 + \frac{1}{2} e^{-2 i \pi x} + \frac{1}{2} e^{2 i \pi x} - \frac{2}{9} e^{-4 i \pi x} - \frac{2}{9} e^{4 i \pi x} + \frac{1}{8} e^{-6 i \pi x} + \frac{1}{8} e^{6 i \pi x} - \frac{2}{25} e^{-8 i \pi x} - \frac{2}{25} e^{8 i \pi x}$



```

gen[l_] :=
Module[
{tone = synthesize[l]},
Column@{Plot[tone, {x, -1, 1}], sound = Sound@musictemplate[
List /@ (CMajor /@ Round@ (
 $\frac{1}{10} * \text{Array}[\text{tone} /. x \rightarrow (\# * 3 / 20 \text{ Pi}) \&, 200]$ 
)),
0.3, "Piano"]}]
]
gen[{1, 1 I, 0, -3, 1, 0, 2 - 3 I, 0, 0, 0, 0, 8, 1, 1, 1, 1, 1, 1, 1, 1, 1}]

```



```
Export["E:\\MyWorkShop\\midi\\demo4.mid", sound]
```

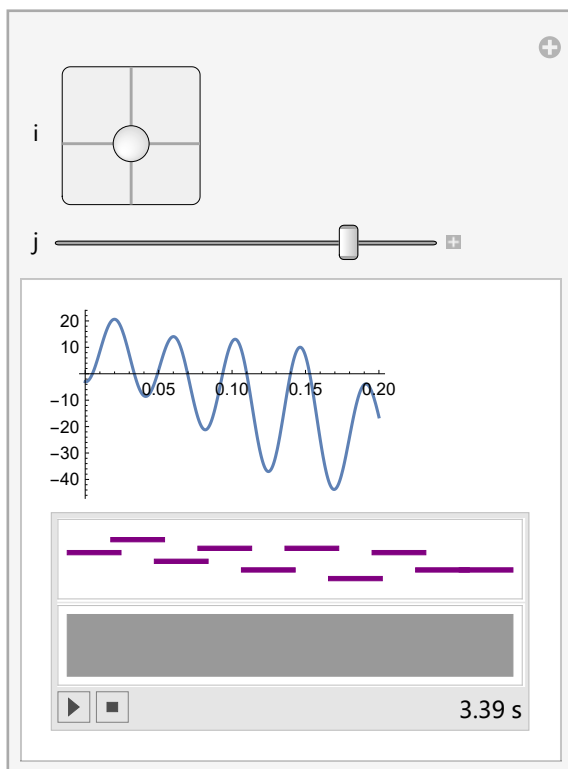
```
E:\\MyWorkShop\\midi\\demo4.mid
```

## Further demonstrate this idea (with manipulation)

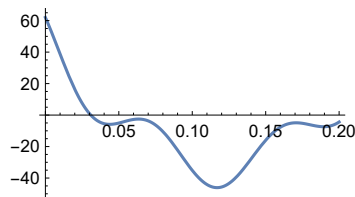
In[254]:=

```
manitone[tone_] := musictemplate[
  List /@ (CMajor /@ Round@ (1/10 * Array[tone /. x -> (# * 1/20) &, 10, -5])),
  0.3, "Piano"]
toneplay[tone_] := Sound[]
longgen[l_] :=
Module[
  {tone = synthesize[l]},
  Column@{Plot[tone, {x, 0, 1/5}], sound = Sound@musictemplate[
    List /@ (CMajor /@ Round@ (1/10 * Array[tone /. x -> (# * 1/50) &, 10, 0])),
    0.3, "Piano"]}]
]
```

**Manipulate[**  
 longgen[{5, 20 I, 3 i[[1]] + 2 i[[2]] I, 0, 0.1, 0, 0, -0.2 I, 0, 0, 0, 0, 0, 0, 0,  
 0, 0, -2, 2 I, 3 - 2 I, 0, 0, 0, 4 - j}], {i, {-10, -10}}, {j, -20, 20}]



```
longgen[{0, 0, 0, 0, 20, 0, 8 - I, 9 I, 0, 0, 3, 10 I}]
```



## A more complex melody(with 2 instruments)

```
In[260]:= long[tone_, nu_] :=
  List /@
    (CMajor /@ Round@ (1/10 * Array[synthesize[tone] /. x -> (nu * 1/5 + # * 1/50) &, 10, 0]))
```

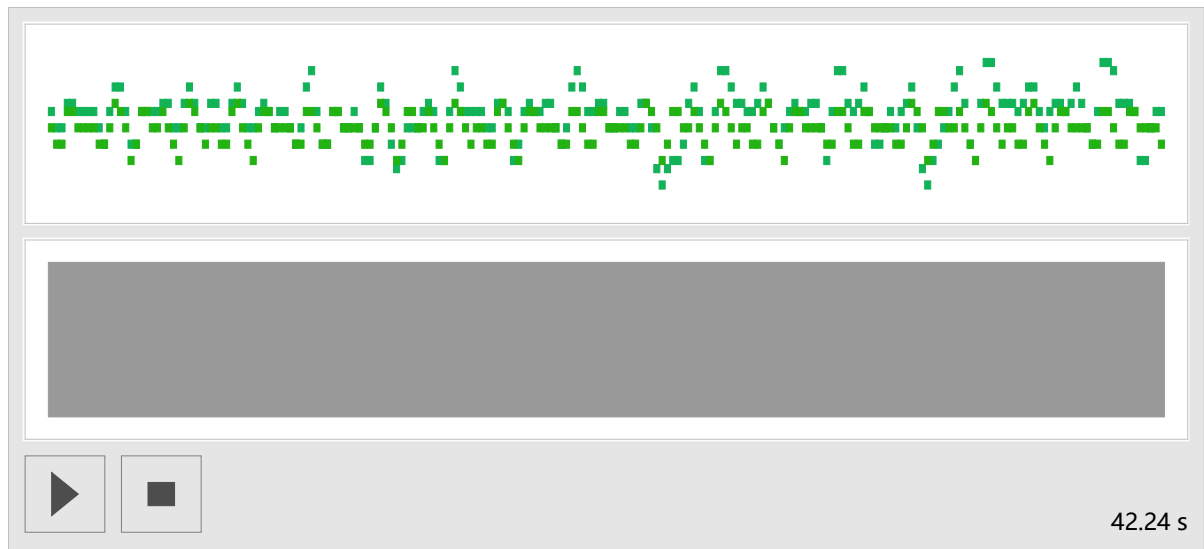
```
In[261]:= Clear[mytone, ftone];
mytone = {0, 0, 0, 0, 0, 0, -I, 4 I, 0, 0, 0, 5 I};
mymusic = long[mytone, 0];
ftone = mytone;
ftone[[1]] = ftone[[1]] - 5;
music1 = long[ftone, 0];
Do[
  Modifier = RandomComplex[{-1 - 1 I, 1 + 1 I}, Length[mytone]];
  mytone = mytone + Modifier;
  mymusic = mymusic ~Join~ long[mytone, i * 1];
  music1 = music1 ~Join~ long[ftone, i * 1];
  , {i, 1, 20}]

Column@{mytone}

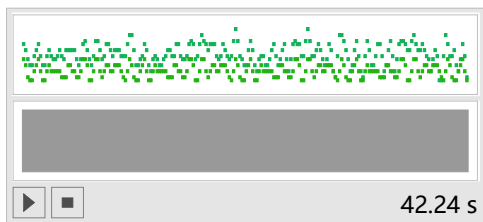
{10.0015 - 2.08178 i, -4.90085 + 2.10142 i, 0.536549 - 0.655445 i,
 0.0841304 - 3.34214 i, 0.902162 + 4.37482 i, 1.85177 + 5.99931 i,
 0.719863 + 1.67582 i, 3.04975 - 0.0113548 i, 2.90536 - 0.844651 i,
 6.63382 - 0.690041 i, 4.26106 + 0.809918 i, 0.714389 + 1.02101 i,
 -1.72149 + 5.47014 i, 1.89154 - 2.66254 i, -0.515178 - 3.05572 i}
```



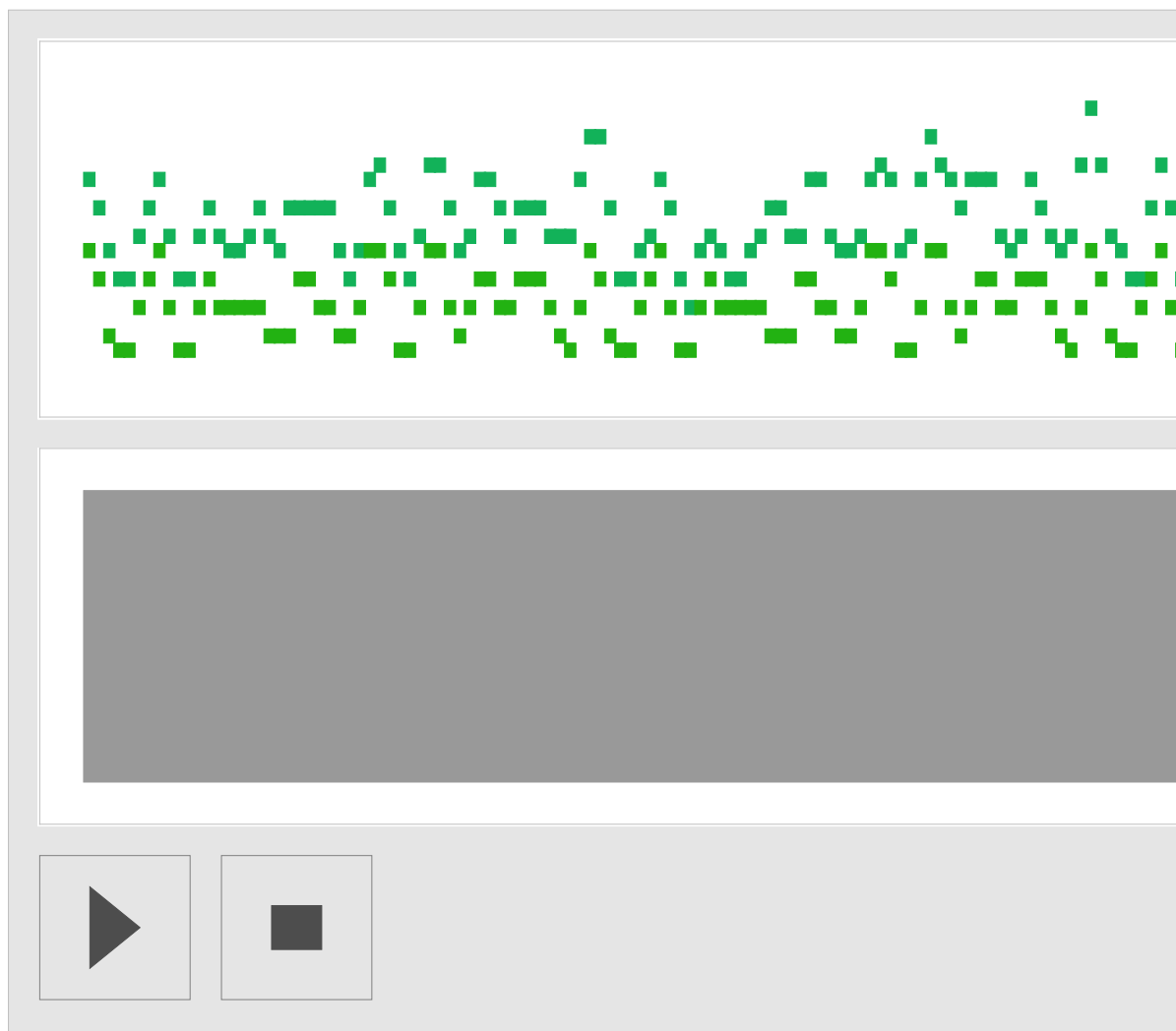
```
Sound[musictemplate[mymusic, 0.2, "Guitar"] ~  
  Join~musictemplate[music1, 0.2, "ElectricBass"]]
```



```
Sound[musictemplate[mymusic, 0.2, "Guitar"] ~  
  Join~musictemplate[music1, 0.2, "ElectricBass"]]
```



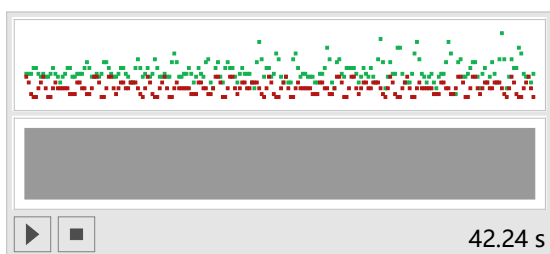
```
demo5##### = Sound[musictemplate[mymusic, 0.2, "Guitar"] ~
Join~musictemplate[music1, 0.2, "ElectricBass"]]
```



```
Export["E:\\MyWorkShop\\midi\\demo5#####.mid", demo5#####]
```

```
E:\\MyWorkShop\\midi\\demo5#####.mid
```

```
demo5#### = Sound[musictemplate[mymusic, 0.2, "Viola"] ~
Join~musictemplate[music1, 0.2, "Trumpet"]]
```



```
long[{0, 1, 1}, 1]
```

```
long[{0, 1, 1}, 1]
```