

ostinato and interrupt

for guitar, unpitched noises, and sustained pitched tones

michael winter (mexico city, mx; 2017)
para fatima

general remarks

The piece consists of a relatively slow, floating ostinato comprised of the following three note cells within which there is a descending bassline (given by the bottom note in each column):

f [#] −14¢	f [#] −14¢	f [#] −14¢	d	d	d
d	d	d	a	a	a
c	b	a [#]	f	e	d

*Note that the f[#] is flat 14 cents, or hundredths of a tempered semitone, from the tempered f[#] in twelve-tone equal temperament as it is a 5/4 major third above the d.

The ostinato is occasionally interrupted by a strictly metered, more rapid sequence of sounds derived from and accompanying open and muted strings as well as natural harmonics played on the guitar.

The interruptions are at a tempo of precisely 75 beats per minute with four 16th notes to the beat (such that the 16th note is always 0.2 seconds). The piece is designed such that the interruptions always start at a second marking to enable coordination and the ability to set the correct tempo by, for example, counting 60 beats per minute (one beat per second) with five 16ths to the beat and then making the metric modulation to 75 beats per minute by simply changing the beat from five 16ths to four 16ths; i.e. changing the tempo while keeping the atomic unit of a 16th note the same at 0.2 seconds. A number below a stem indicates a change in the duration of the following notes: 1 equals 0.2 seconds, 2 equals 0.4 seconds (or two times the atomic duration), or 3 equals 0.6 seconds (or three times the atomic duration).

guitar

The piece is essentially a guitar piece that can be accompanied by computer (with custom software written in SuperCollider) and / or live performers. In the following sections, each accompanying element is detailed with an explanation of what is occurring in the computer program and how it can be substituted or complemented by live performers.

The notation is given in tablature-form where each line represents one of the strings of the guitar from high to low. A filled-notehead indicates an open string. A number indicates a stopped string at the fret of the given number. A diamond-notehead indicates a natural harmonic (chosen arbitrarily / randomly). And an x-notehead indicates a muted string.

The guitar is tuned in an open d tuning as follows: VI) e down to d, V) a, IV) d, III) g down to f[#] −14¢, II) b up to c −31¢, and I) e down to d. Strings IV and III can be tuned from the 5th and 7th harmonic of string VI, respectively.

The ostinato sections should be interpreted freely. The guitar enters 12 seconds into the piece. The given notes, written proportionally with 12 seconds per system, should serve more as a general guideline. The guitarist is free to embellish ad lib such that initially, embellishments are rather infrequent resulting in a relatively sparse texture, and as time progresses, the general density of the ostinato increases by the aid of more and more embellishments. The embellishments should remain within the harmonic world of the ostinato.

The computer program allows the guitarist to use sampled recordings of a guitar which will accurately produce what is written. The guitarist may choose to play with a sample-based realization and / or with other guitarists; e.g. by dividing the ostinato and interrupt sections among multiple performers and / or the computer realization. Note that the electronic accompaniment makes decisions at every point indicated by a notehead in the ostinato sections (explained in detail below).

The interruption sections are clearly delimited from the ostinato by double bar lines and because they are strictly metered. For both the ostinato and interrupt sections, tones should be allowed to decay as long as possible.

unpitched noises (noise fields and percussion)

The ostinato and interrupt sections are further distinguished by noise fields. The computer program simply oscillates between brown noise for the ostinato sections and white noise for the interruptions. These sounds can be replaced and / or complemented by live performers choosing two distinct noises for the ostinato and interrupt sections, respectively. The level of the noise fields should be relatively unobtrusive and situated at a level well below the guitar. The noise fields fade out over 10 seconds after the final note of the guitar.

In the interrupt sections, unpitched short percussion sounds double the muted strings of the guitar (indicated by x-noteheads) using a distinct instrument for every line of the staff. While a live performer is highly preferred, the computer program allows this part to be realized using sampled recordings. The score includes a percussion part with all non x-noteheads grayed out.

sustained tones (harmonic flickering, sine beating, and interrupt highlights)

The bassline of the ostinato can be optionally highlighted by what can be described as a ‘harmonic flickering’ where harmonics of the bassline flit in and out with various degrees of pitch definition. This is produced in the computer program by a simple waveshaping technique where a buffer is intermittently filled with bursts of noise and played back such that position of the player resets to the beginning of the buffer at a rate equivalent to the frequency of the current bass note in the ostinato. These sounds can be replaced and / or complemented by live performers playing swelled tones and occasionally mimicking the synthesis process on harmonics (including the fundamental) of the current bass note. Note that at each ostinato note, the computer can either turn on or off the flickering. The flickering nature of the sound itself is created by randomly gating the buffer player at very short intervals and the random intervals in which the buffer is refilled. If the flickering is realized by live performers, a similar decision making process should be coordinated. In the notation, brackets around the notes indicate when the bassline progresses. The harmonic flickering should be relatively unobtrusive and situated at a level well below the guitar.

The ostinato and interrupt sections are also distinguished by the beating of two low tones between 0.5 and 3 hertz apart centered around d for the ostinato sections and a fourth below, a, for the interrupt sections. This is achieved using sine tones in the computer program, but may be replaced by live performers playing low instruments. Note that the sine tones, and thus preferably any instruments that replace the sine tones, are actually centered an octave below the VI string of the guitar in the ostinato sections and 2 octaves below the V string of the guitar in the interrupt sections. In the ostinato sections, the computer determines whether or not to change the rate of beating on the onset of every note. This can easily be reproduced by two players such that one remains constant while the other changes occasionally. The rate of beating remains constant throughout each interrupt section. The beating should be at a level such that it provides a clear and present foundation while not overwhelming any of the other sounds.

In the interrupt sections, every open string and natural harmonic played on guitar is accompanied / highlighted by sustained tones that are octave equivalents of the fundamental or harmonics above the fundamental, respectively. For the latter, any harmonic greater than 2 can be chosen randomly / arbitrarily. The computer uses sine tones with amplitudes equivalent to 1 divided by the harmonic number. It is encouraged that live performers complement instead of replace the computer generated tones such that each player interprets one line of the score realizing filled-noteheads as octave equivalents of the given open string of the guitar and diamond-noteheads as octave equivalents of natural harmonics of the given string. If possible, as with the synthesized tones, performers should try to play these tones at a level indirectly proportional to the harmonic number. If a filled-notehead is repeated, the tone may be rearticulated or played in a different octave. A different harmonic can be chosen every time a diamond-notehead occurs. An x-notehead indicates to stop the currently sounding tone. If a sound has not been stopped before a transition back into the ostinato section, the tone fades out over a few seconds slightly overlapping into the following ostinato section. The score includes 6 ensemble parts where all but the relevant stafflines and noteheads are grayed out. The same performers can realize both the harmonic flickering of the ostinato and the interrupt highlights. To facilitate performances with less than 6 performers realizing the highlights, the parts may be divided among performers and the computer program (which allows muting individual parts).

Note that where octave equivalents of harmonics are played (such as in the harmonic flickering and the interrupt highlights), several pitches deviate from the nearest pitch in twelve tone equal-temperament. Below, the first 6 unique pitch classes (based on primes) of the harmonic series on d are listed with a cents deviation from the nearest pitch in twelve-tone equal-temperament. These pitches can be transposed accordingly for all other cases.

d	a +2¢	f [♯] -14¢	c -31¢	g [♯] -49¢	a [♯] +40¢
2	3	5	7	11	13

SuperCollider program structure

The structure of the application is hopefully straightforward and does not warrant much explanation. The application launches a graphical user interface (gui) that controls each element explained in an environment similar to a digital audio workstation (daw). Each element is played back from a multichannel soundfile. A timer in minutes:seconds and a visual metronome at 75 beats per minute (for the interrupt sections) is provided for coordination. Images of each tab of the gui and the directory structure of the application resources are provided below. The channels of the soundfile (24 in total) correspond to the faders from left to right.

To launch the application, execute `ostinato_and_interrupt_main.scd` in SuperCollider after booting the server (on linux, this is achieved by pressing `cmd+enter` with the cursor anywhere within the code block).

The “generate” button regenerates the piece. By default it will generate the original version included with this score, however the random seed can be changed if someone is so bold as to try to create a new version. Note that the application was written to create the given version, but should hopefully function properly when generating new version even though it has not been extensively tested. Regenerating the piece creates / replaces several files: most importantly `ostinato_and_interrupt.wav` which is needed for playback and should open in most daws (tested with Audacity). It also regenerates the Lilypond files which can be rendered and engraved using Lilypond.

For the generation function to work properly, the application requires that single samples be placed in the `samples/` folder within the directory tree given on the following page. Ideally, there should be several samples for each sound. While the application does not adhere to any naming conventions for the sample files, changing the names of the folders will break the application. The number prefix of each folder applies to the string / part number. With exception of the `strings_harmonics/` folder, the samples within a folder should be different versions of the same sound.

ostinato_bass/: each folder contains samples of the given bass note of the ostinato from highest (1) to lowest (6) (the descending bassline is described in the beginning of the instructions).

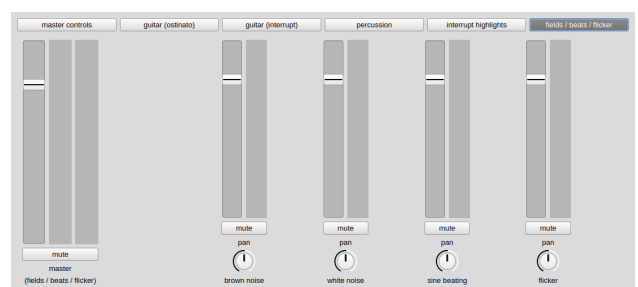
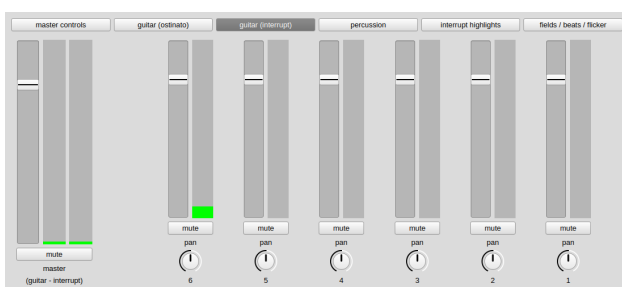
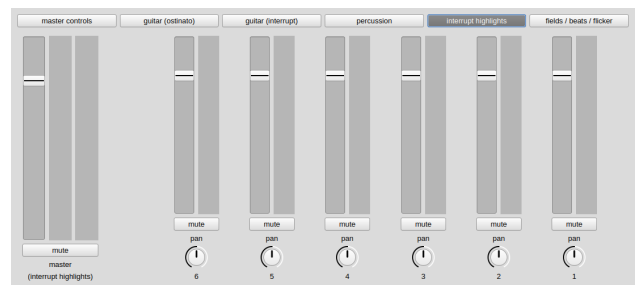
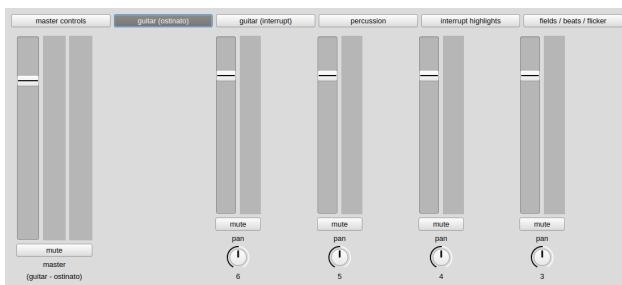
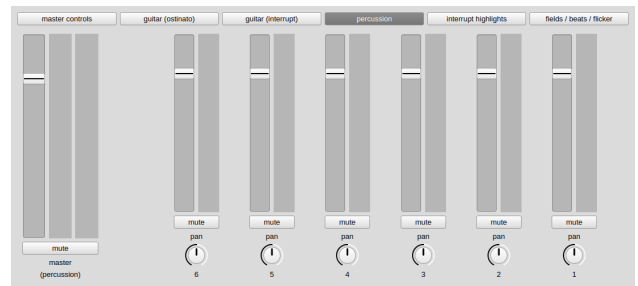
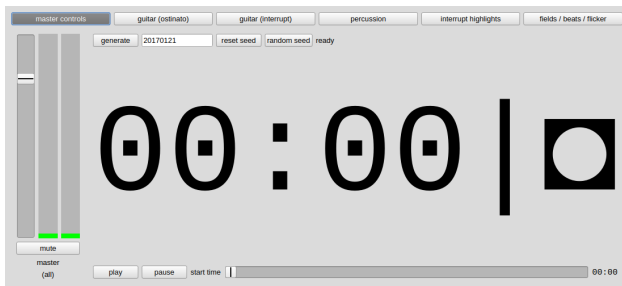
strings_open/: each folder contains samples of the given open string from highest (1) to lowest (6).

strings_open/: each folder contains samples of different natural harmonics of the given string from highest (1) to lowest (6). These are the only folders of the guitar samples that will contain sounds of different pitches. Note that if several samples of each harmonic are recorded, then there must be the same number of each harmonic in each folder; e.g. two samples of the 2nd harmonic, two samples of the 3rd harmonic, etc.

strings_muted/: each folder contains samples of the given string muted from highest (1) to lowest (6).

percussion/: each folder contains samples of one of the six different percussion instruments.

The primary source code for the application is appended at the end of this score and can be can be downloaded from a git repository at https://www.github.com/mwinter80/ostinato_and_interrupt. The whole package including all audio file resources is available upon request or can be downloaded from http://www.unboundedpress.org/code_releases/ostinato_and_interrupt_source.zip. Note that the package comes with a pregenerated version of the piece (the multichannel soundfile and all the Lilypond files) using an included sample set. This original sample set (recorded in August of 2017) was not recorded in ideal conditions and was used simply to audition the piece. The generation of this document (using LaTeX) and the musical parts (in Lilypond) contain version dates in order to help track changes and the git repository will also detail commit changes. The piece was written using SuperCollider version 3.8.0 and Lilypond version 2.18.2.



```

ostinato_and_interrupt_source/
├── supercollider/
│   ├── ostinato_and_interrupt_main.scd
│   ├── ostinato_and_interrupt_generator_synthdef.scd
│   ├── ostinato_and_interrupt_nrt_generator_function.scd
│   ├── ostinato_and_interrupt_lilypond_generator_function.scd
│   ├── ostinato_and_interrupt_player_synthdef.scd
│   ├── ostinato_and_interrupt_gui_generator_function.scd
│   └── gen_data_resources/
│       ├── ostinato_and_interrupt_osc
│       └── ostinato_and_interrupt_data.wav
├── audio/
│   └── ostinato_and_interrupt.wav
├── lilypond/
│   ├── ostinato_and_interrupt_lilypond_score_template.ly
│   ├── ostinato_and_interrupt_lilypond_guitar_part.ly
│   ├── ostinato_and_interrupt_lilypond_percussion_part.ly
│   ├── ostinato_and_interrupt_lilypond_ensemble_part_1.ly
│   ├── ostinato_and_interrupt_lilypond_ensemble_part_2.ly
│   ├── ostinato_and_interrupt_lilypond_ensemble_part_3.ly
│   ├── ostinato_and_interrupt_lilypond_ensemble_part_4.ly
│   ├── ostinato_and_interrupt_lilypond_ensemble_part_5.ly
│   └── ostinato_and_interrupt_lilypond_ensemble_part_6.ly
├── score_pdfs/
│   ├── ostinato_and_interrupt_lilypond_guitar_part.pdf
│   ├── ostinato_and_interrupt_lilypond_percussion_part.pdf
│   ├── ostinato_and_interrupt_lilypond_ensemble_part_1.pdf
│   ├── ostinato_and_interrupt_lilypond_ensemble_part_2.pdf
│   ├── ostinato_and_interrupt_lilypond_ensemble_part_3.pdf
│   ├── ostinato_and_interrupt_lilypond_ensemble_part_5.pdf
│   ├── ostinato_and_interrupt_lilypond_ensemble_part_4.pdf
│   └── ostinato_and_interrupt_lilypond_ensemble_part_6.pdf
└── samples/

```

```

ostinato_and_interrupt_source/
├── samples/
│   ├── ostinato_bass/
│   │   ├── 1_bass/
│   │   ├── 2_bass/
│   │   ├── 3_bass/
│   │   ├── 4_bass/
│   │   ├── 5_bass/
│   │   └── 6_bass/
│   ├── strings_open/
│   │   ├── 1_open/
│   │   ├── 2_open/
│   │   ├── 3_open/
│   │   ├── 4_open/
│   │   ├── 5_open/
│   │   └── 6_open/
│   ├── strings_harmonics/
│   │   ├── 1_harmonic/
│   │   ├── 2_harmonic/
│   │   ├── 3_harmonic/
│   │   ├── 4_harmonic/
│   │   ├── 5_harmonic/
│   │   └── 6_harmonic/
│   ├── strings_muted/
│   │   ├── 1_muted/
│   │   ├── 2_muted/
│   │   ├── 3_muted/
│   │   ├── 4_muted/
│   │   ├── 5_muted/
│   │   └── 6_muted/
│   └── percussion/
│       ├── 1_percussion/
│       ├── 2_percussion/
│       ├── 3_percussion/
│       ├── 4_percussion/
│       ├── 5_percussion/
│       └── 6_percussion/

```

I would like to extend a special thanks to Jose Manuel Alcantara and Alex Bruck for their help during this piece: Jose Manuel for his encouragement and for lending me a guitar; and Alex for his generosity in answering many, many questions and his patience when I just needed to someone to listen as I talked through ideas... this piece was very much shaped by our friendship...

version generated: 2017.12.11

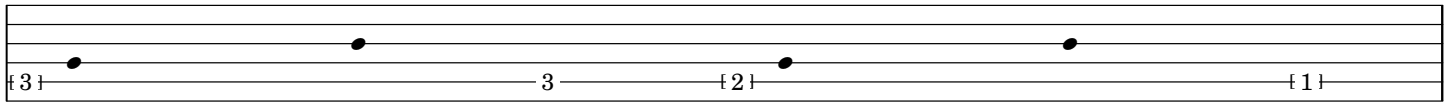
ostinato and interrupt

guitar/all

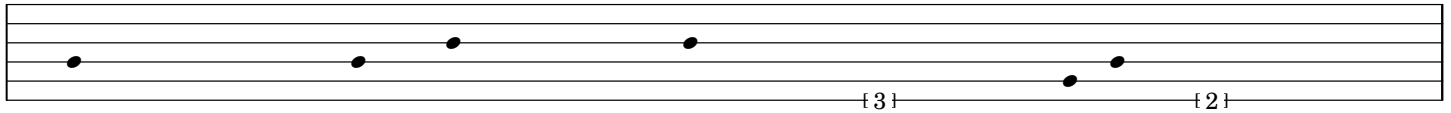
michael winter (mexico city, mx; 2017)

version generated: 2017.08.23

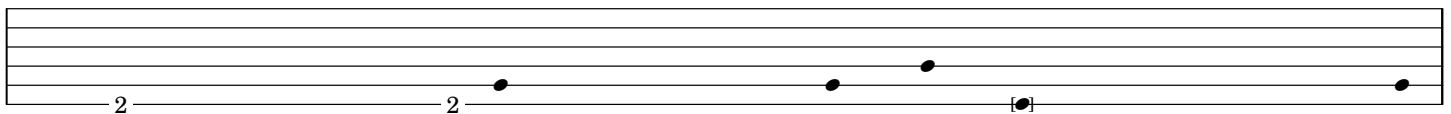
0'12"



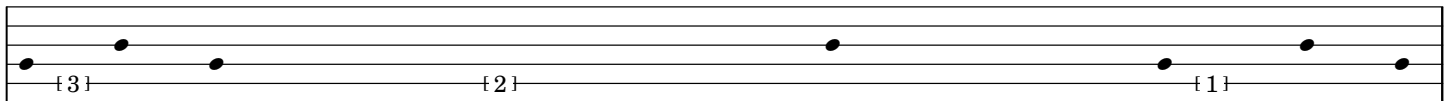
0'24"



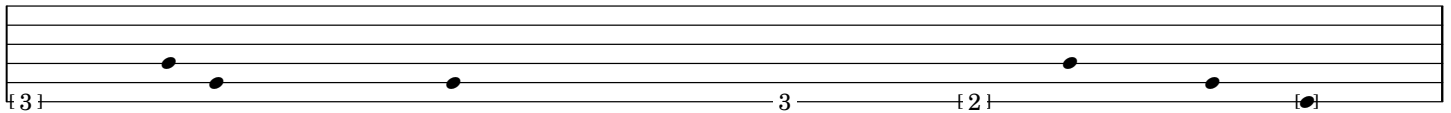
0'36"



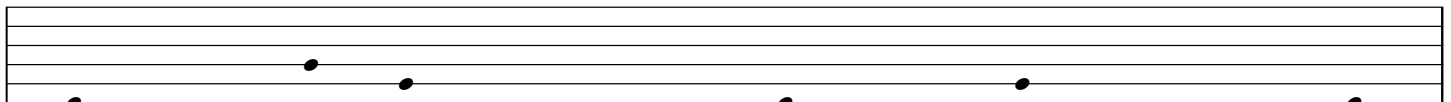
0'48"



1'00"

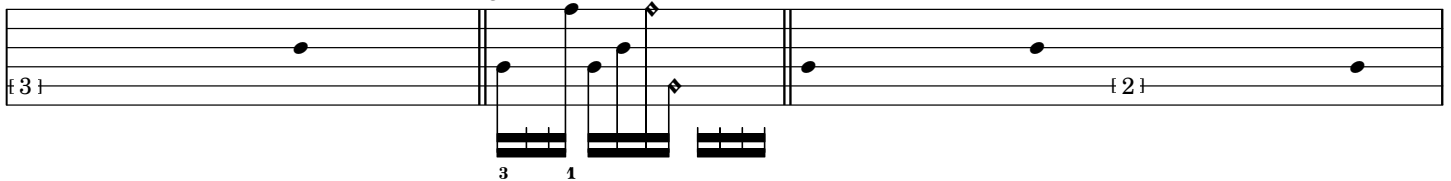


1'12"

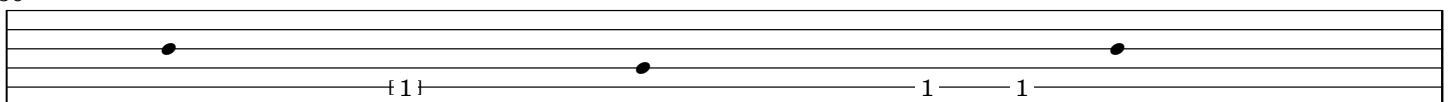


1'24"

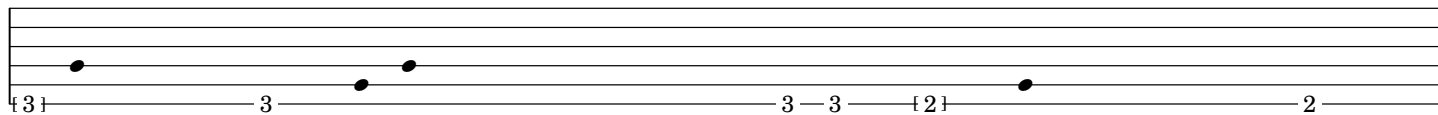
1'28"



1'36"

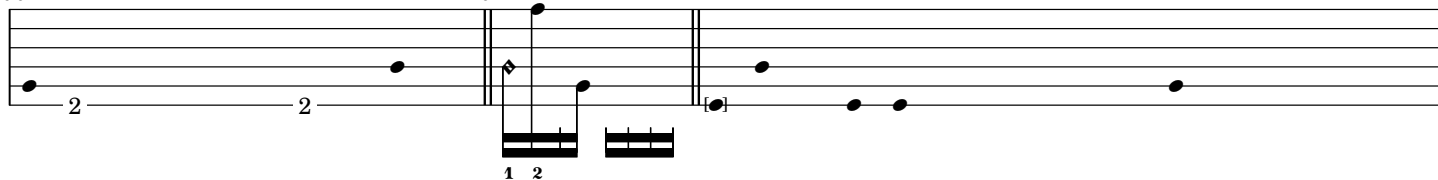


1'48"

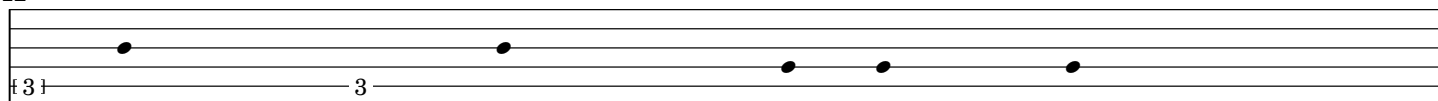


2'00"

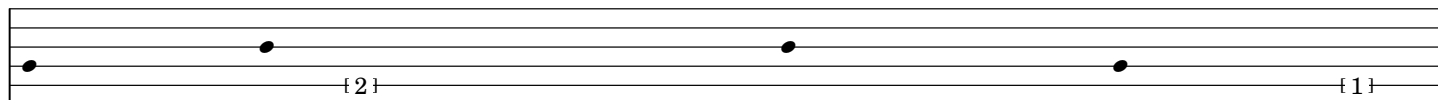
2'04"



2'12"

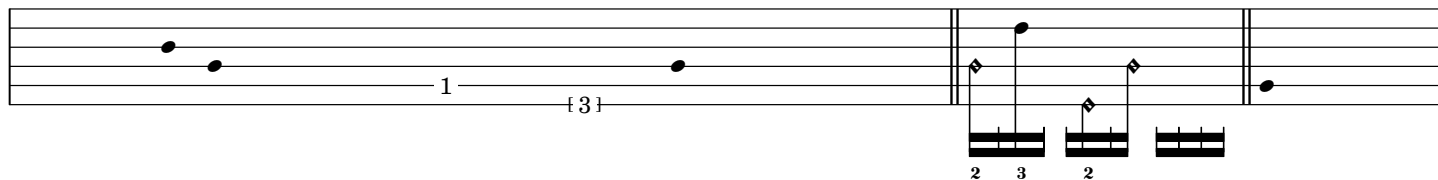


2'24"

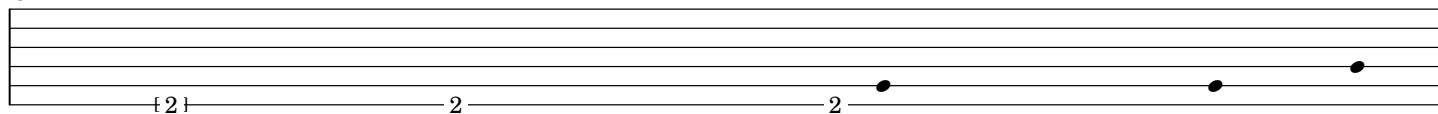


2'36"

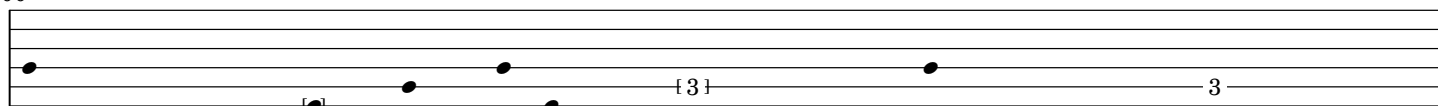
2'44"



2'48"

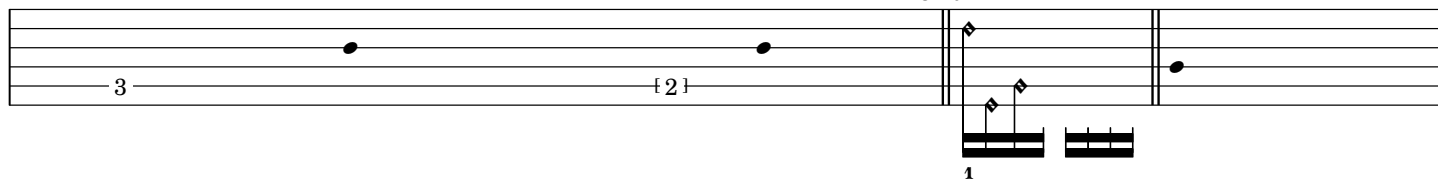


3'00"



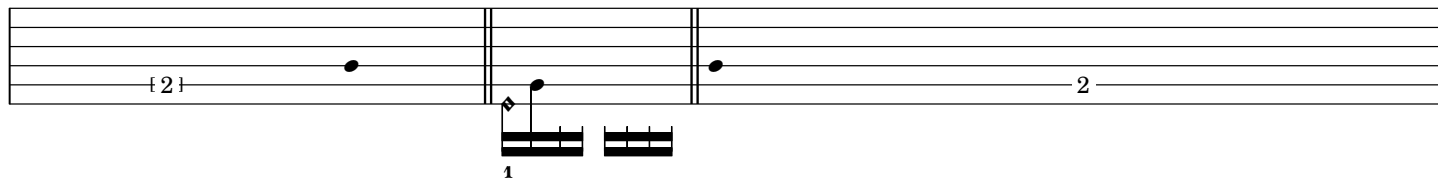
3'12"

3'20"

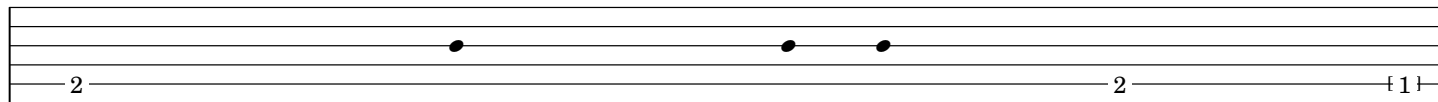


5'00"

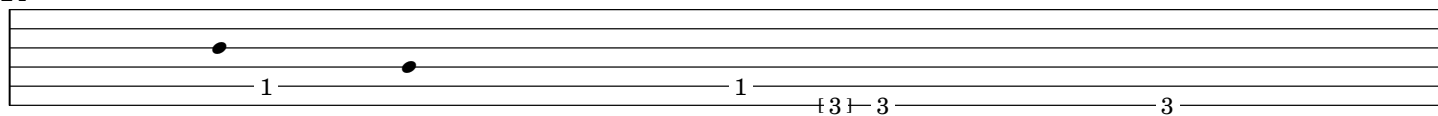
5'04"



5'12"

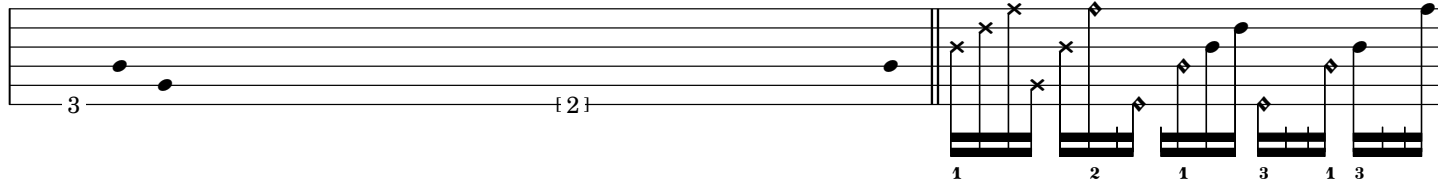


5'24"

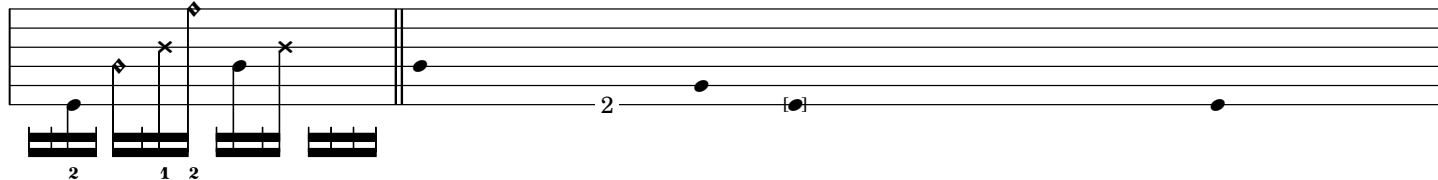


5'36"

5'44"



5'48"

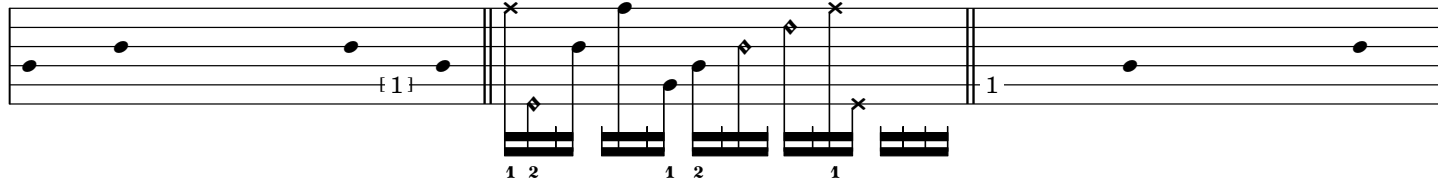


6'00"

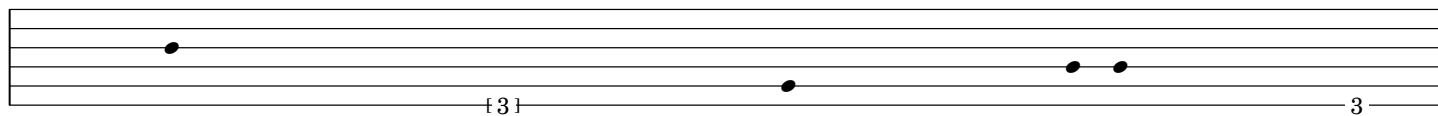


6'12"

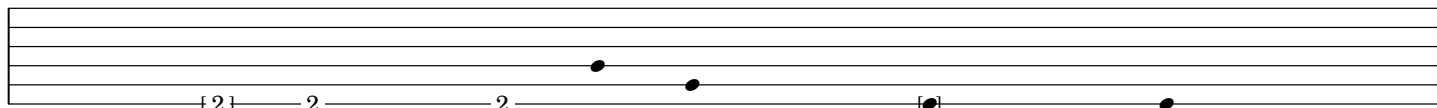
6'16"



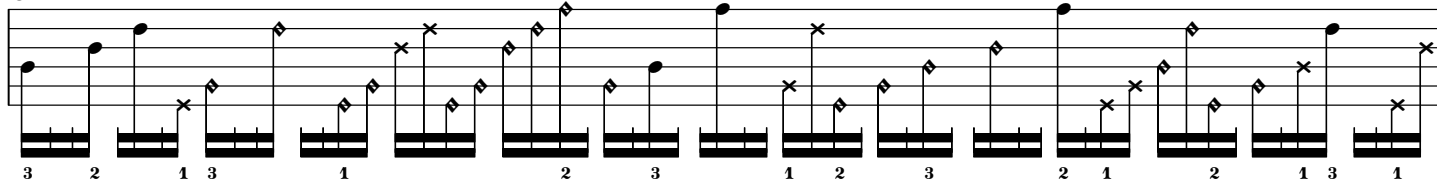
6'24"



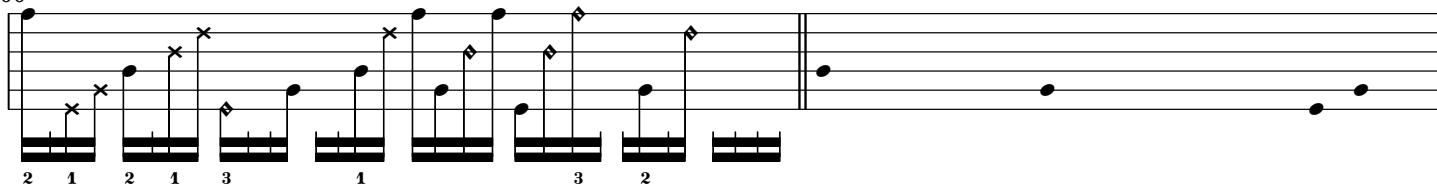
6'36"



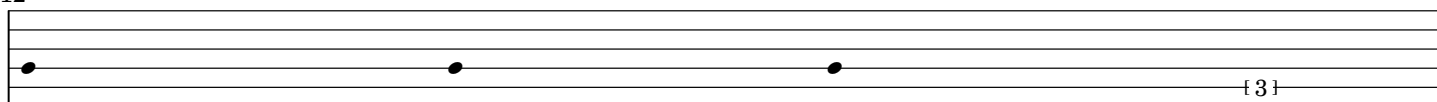
6'48"



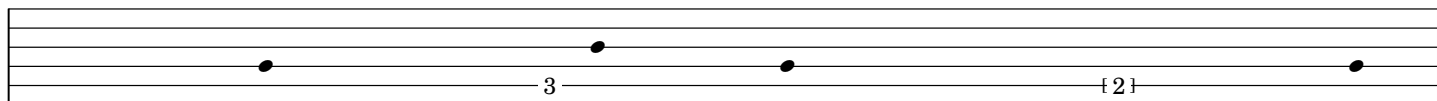
7'00"



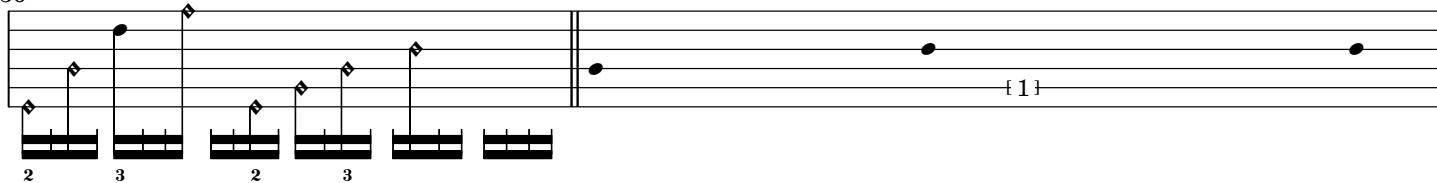
7'12"



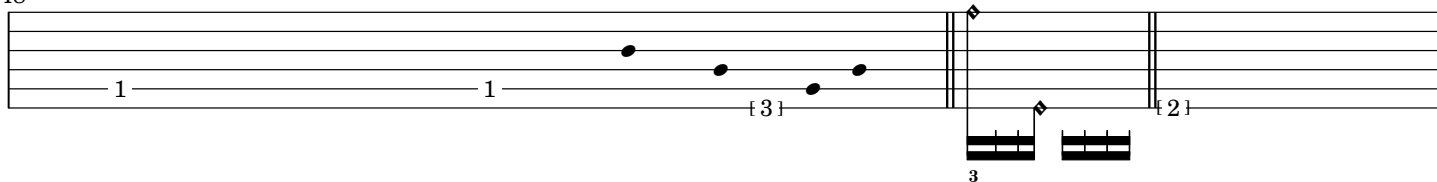
7'24"



7'36"

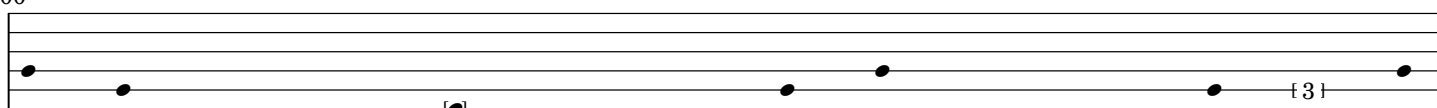


7'48"



7'56"

8'00"



8'12"

Musical notation for 8'12". The staff shows a sequence of notes with fingerings (3, 1, 3, 1, 3, 1) and a triplet of eighth notes. A double bar line is followed by a triplet of eighth notes.

8'24"

Musical notation for 8'24". The staff shows a sequence of notes with a triplet of eighth notes and a double bar line.

8'36"

8'40"

Musical notation for 8'36" and 8'40". The staff shows a sequence of notes with fingerings (1, 2, 1, 3, 1, 2, 1, 2, 1, 3) and a double bar line.

8'48"

Musical notation for 8'48". The staff shows a sequence of notes with a triplet of eighth notes and a double bar line.

9'00"

9'04"

Musical notation for 9'00" and 9'04". The staff shows a sequence of notes with fingerings (1, 3, 1, 2) and a double bar line.

9'12"

Musical notation for 9'12". The staff shows a sequence of notes.

9'24"

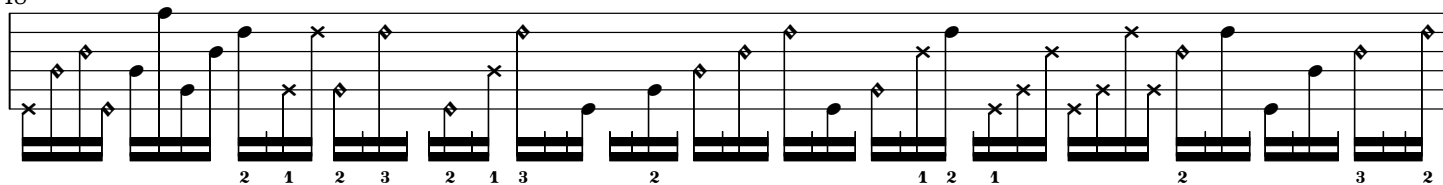
Musical notation for 9'24". The staff shows a sequence of notes.

9'36"

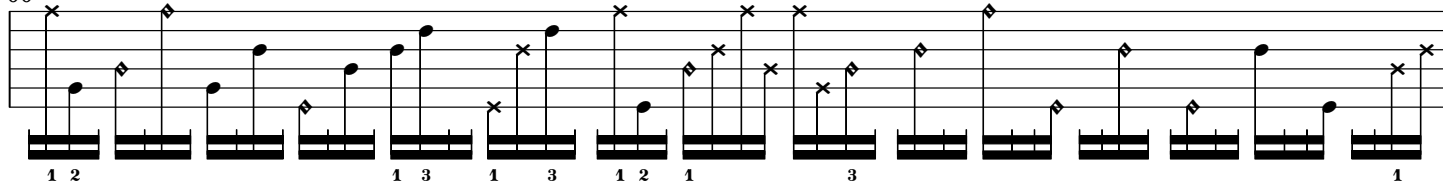
9'44"

Musical notation for 9'36" and 9'44". The staff shows a sequence of notes with fingerings (3, 1, 2, 1) and a double bar line.

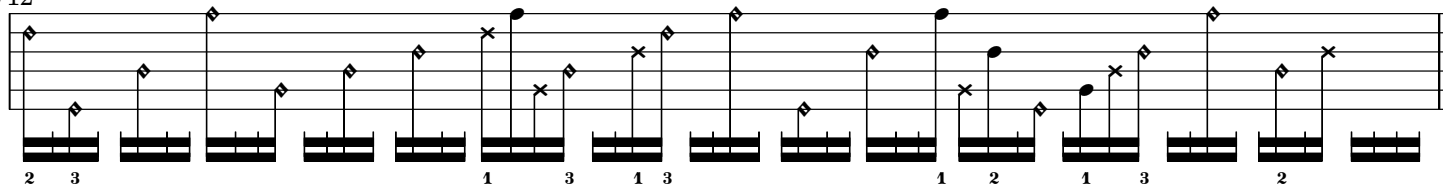
9'48"



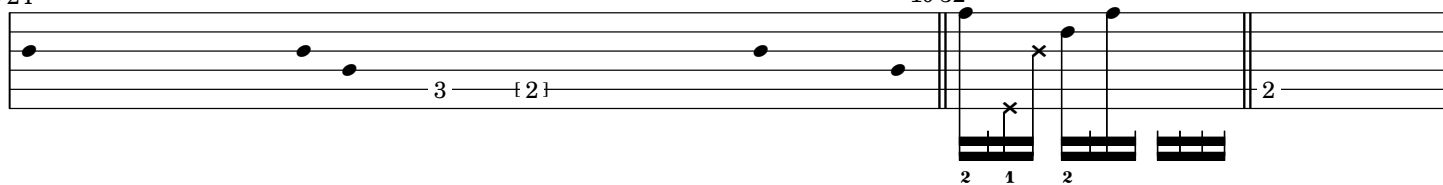
10'00"



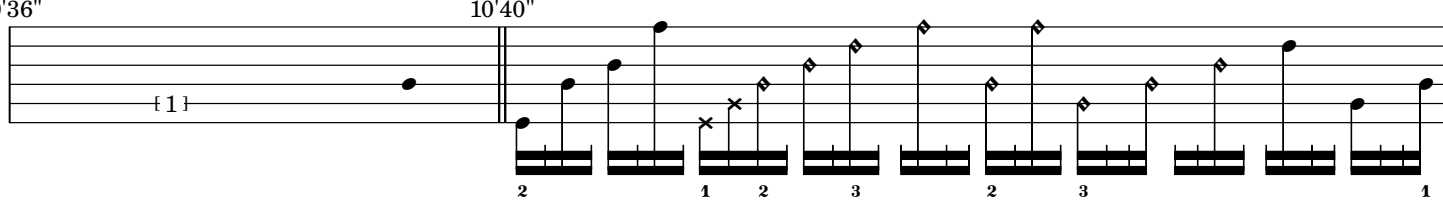
10'12"



10'24"

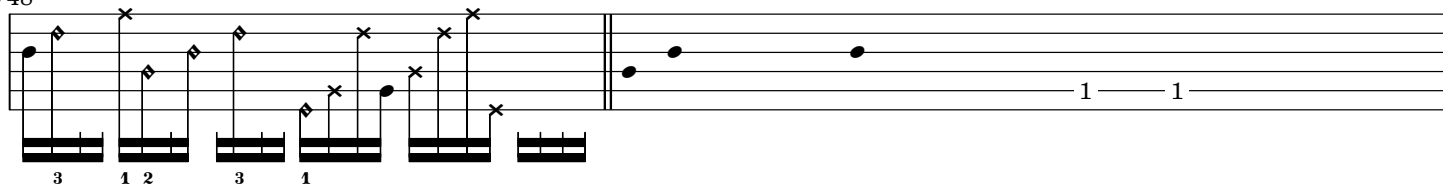


10'36"

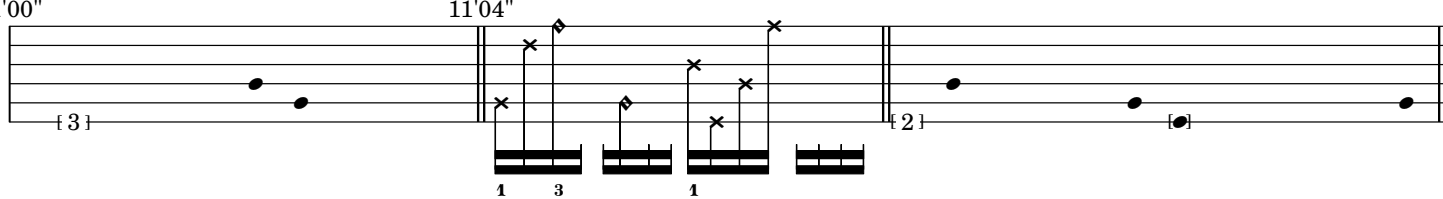


10'40"

10'48"

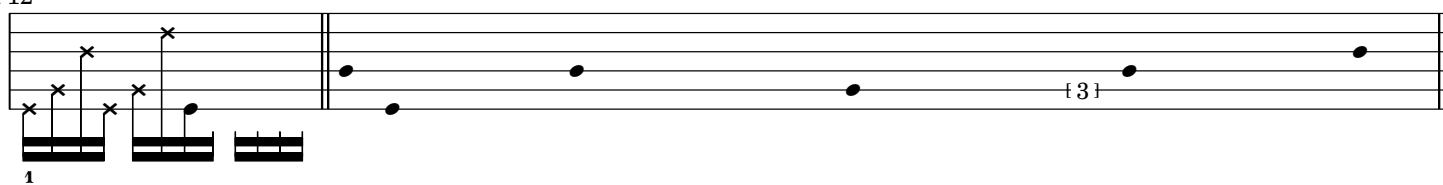


11'00"

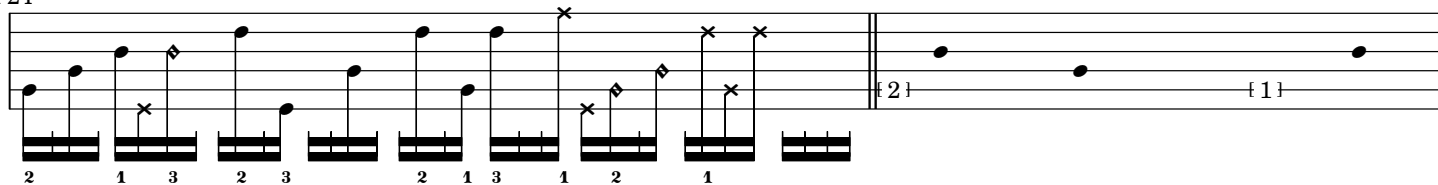


11'04"

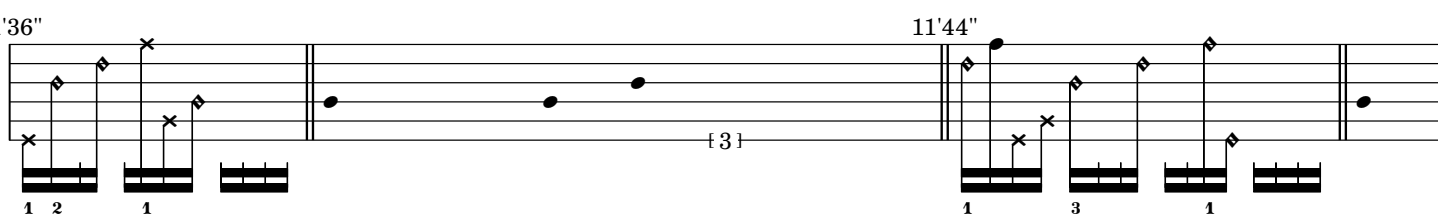
11'12"



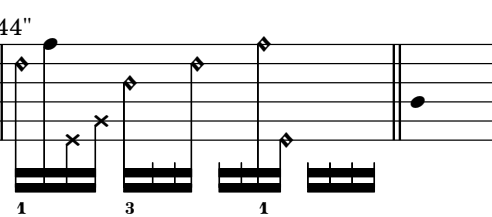
11'24"



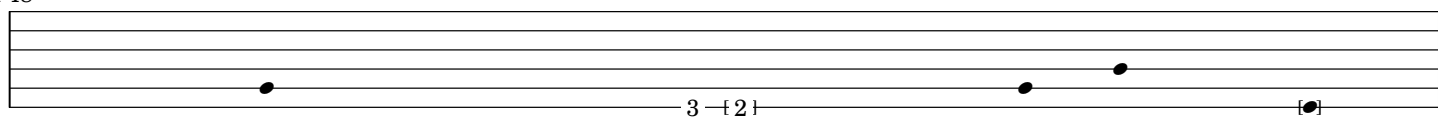
11'36"



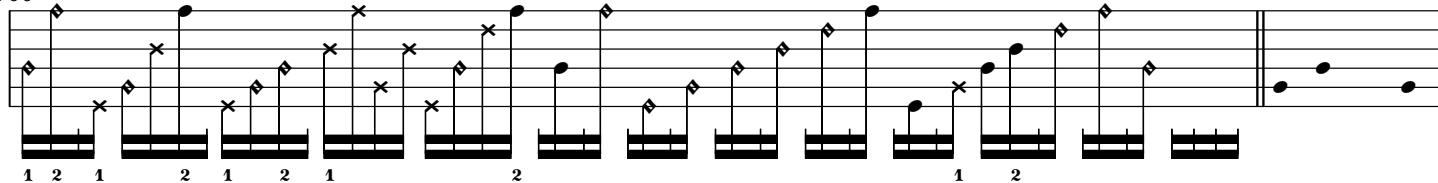
11'44"



11'48"



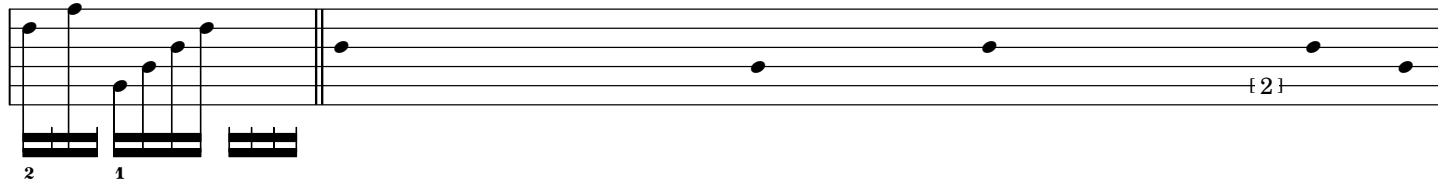
12'00"



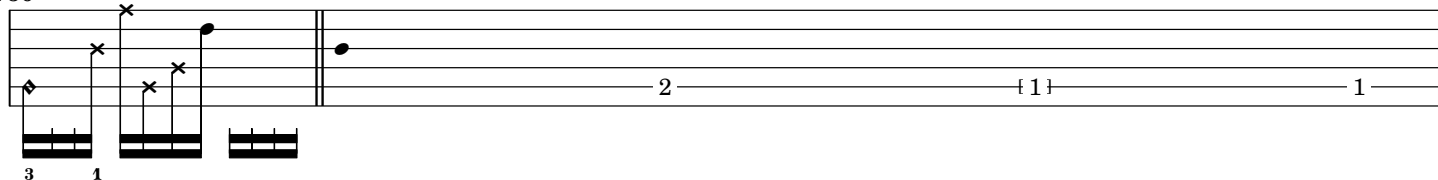
12'12"



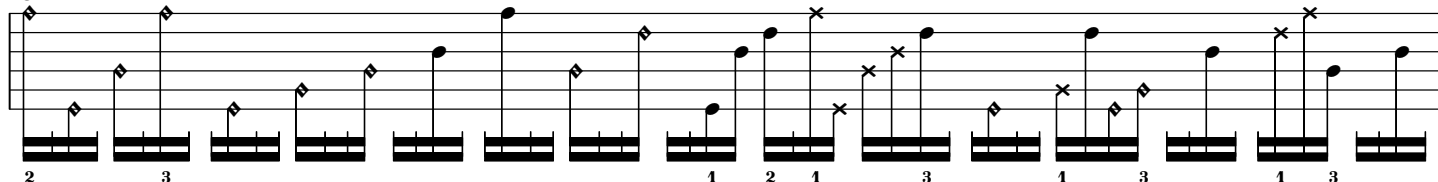
12'24"



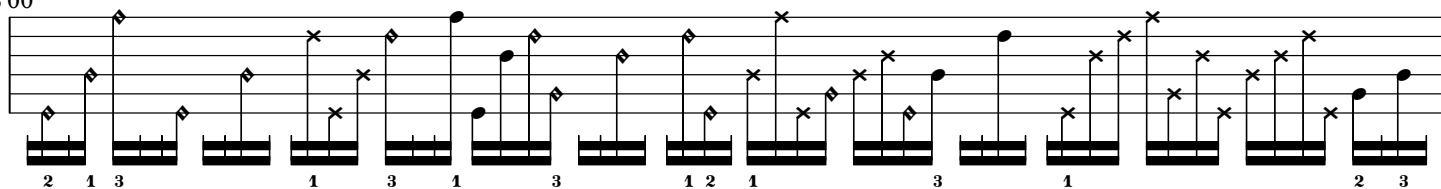
12'36"



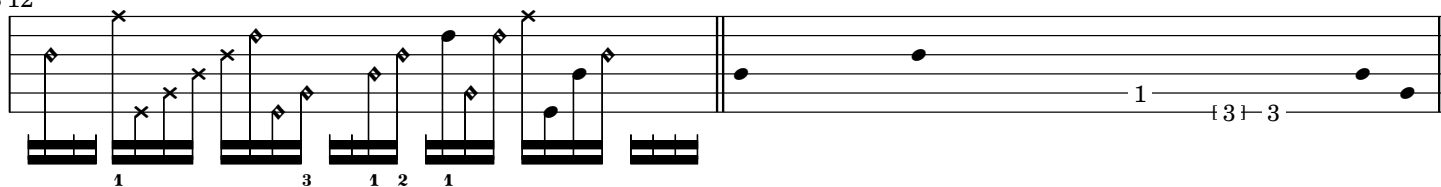
12'48"



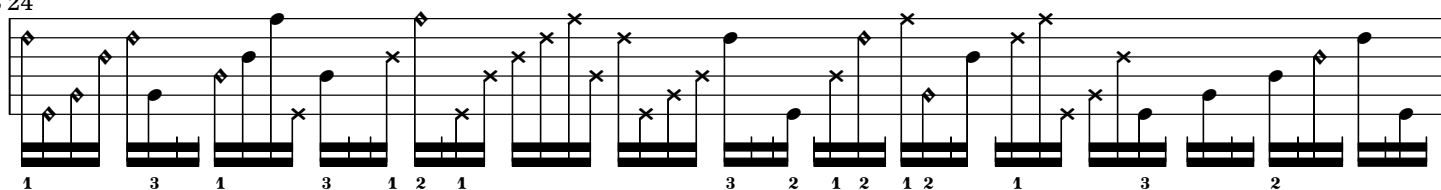
13'00"



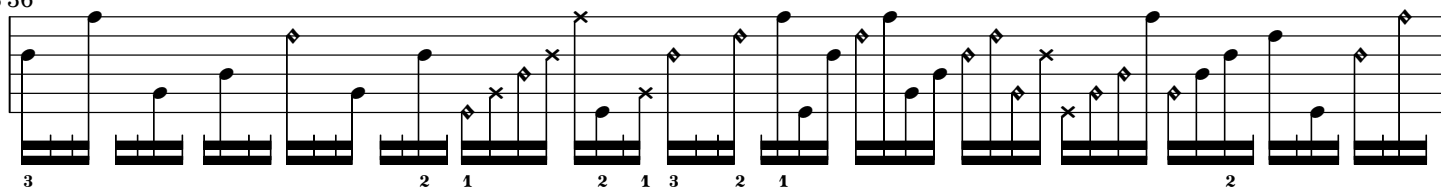
13'12"



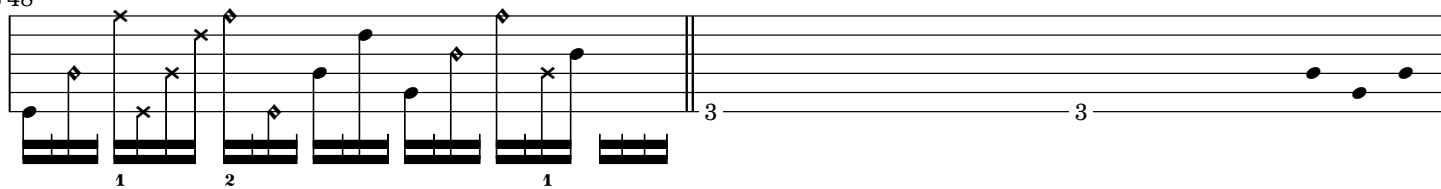
13'24"



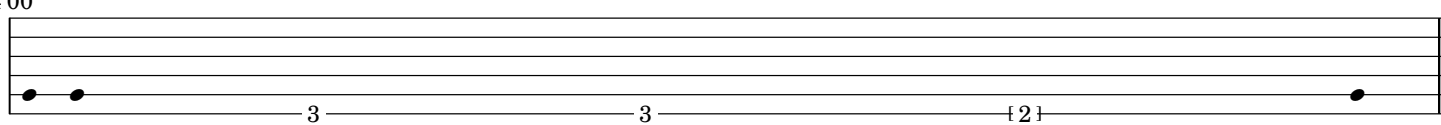
13'36"



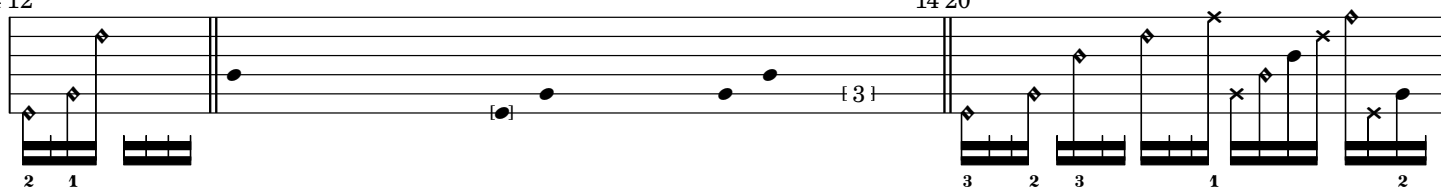
13'48"



14'00"

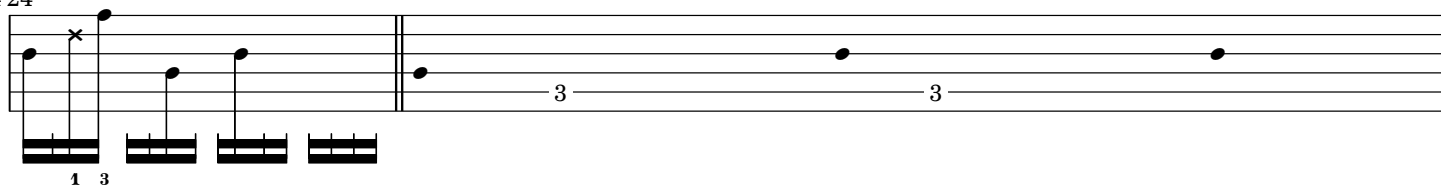


14'12"



14'20"

14'24"



14'36"

Musical notation for 14'36". The staff shows a long rest of 3 measures, followed by a double bar line and a rest of 2 measures. The notation is labeled with a brace and the number 2.

14'44"

14'48"

Musical notation for 14'48". The staff shows a sequence of notes and rests, with fingerings indicated by numbers 1, 2, and 3. The notation is labeled with a brace and the number 2.

15'00"

Musical notation for 15'00". The staff shows a sequence of notes and rests, with fingerings indicated by numbers 1, 2, and 3. The notation is labeled with a brace and the number 2.

15'12"

Musical notation for 15'12". The staff shows a sequence of notes and rests, with fingerings indicated by numbers 1, 2, and 3. The notation is labeled with a brace and the number 2.

15'24"

Musical notation for 15'24". The staff shows a sequence of notes and rests, with fingerings indicated by numbers 1, 2, and 3. The notation is labeled with a brace and the number 3.

15'36"

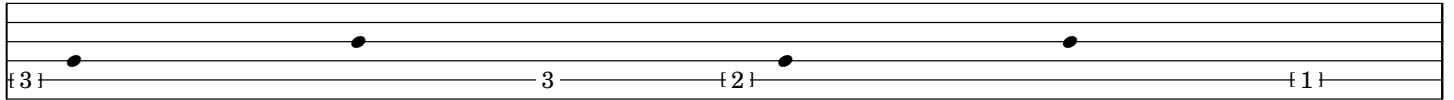
Musical notation for 15'36". The staff shows a sequence of notes and rests, with fingerings indicated by numbers 1, 2, and 3. The notation is labeled with a brace and the number 2.

ostinato and interrupt

michael winter (mexico city, mx; 2017)
version generated: 2017.08.23

percussion

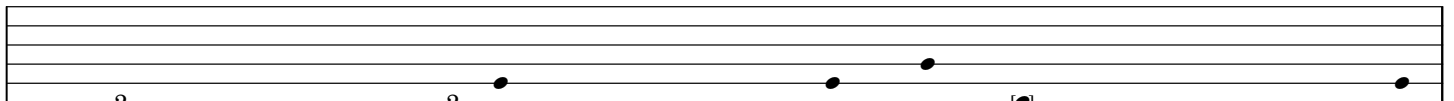
0'12"



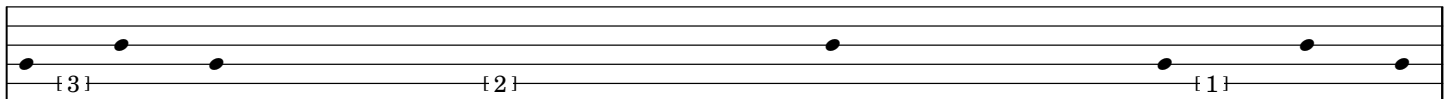
0'24"



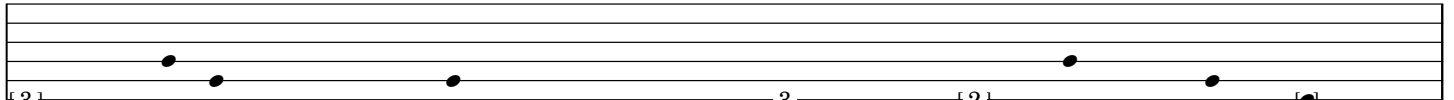
0'36"



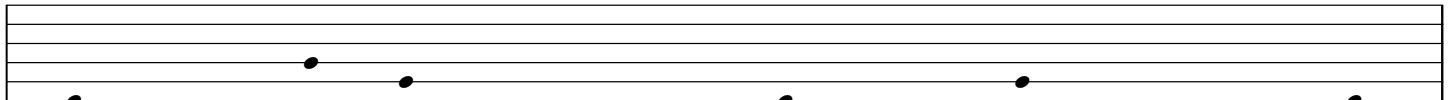
0'48"



1'00"

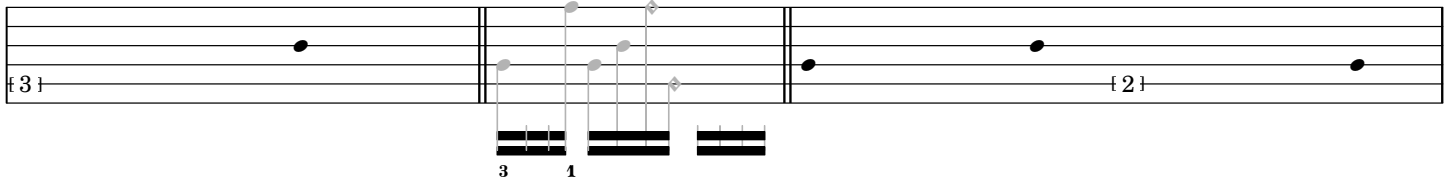


1'12"

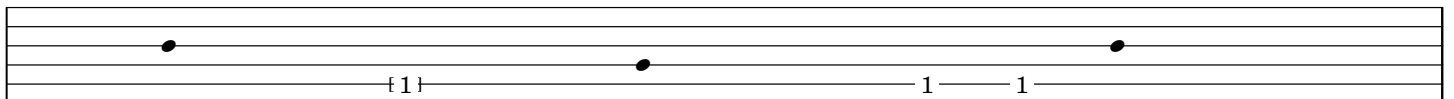


1'24"

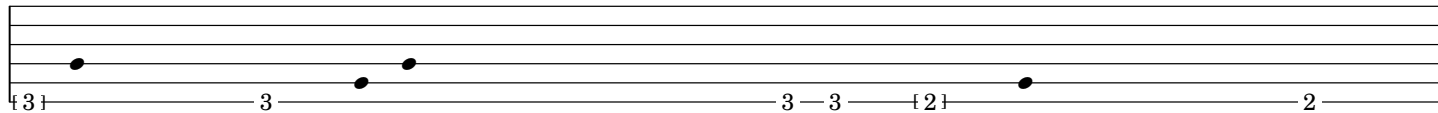
1'28"



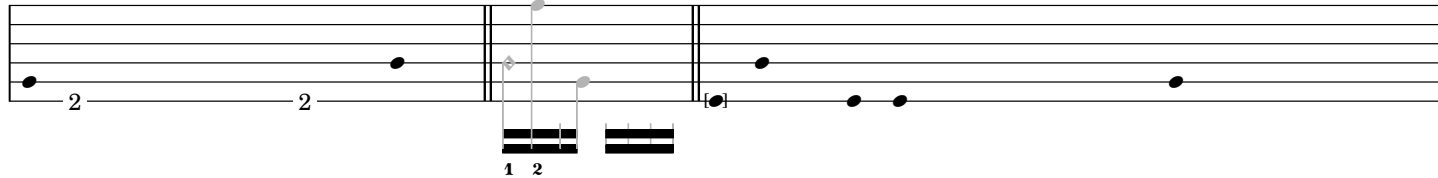
1'36"



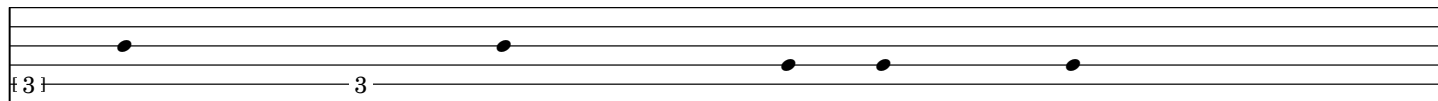
1'48"



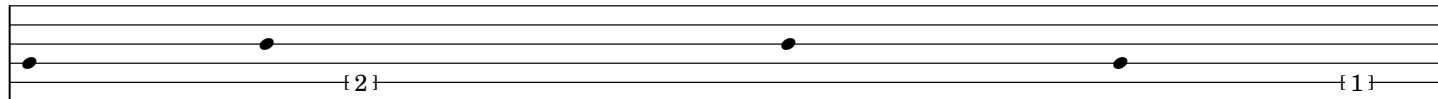
2'00"



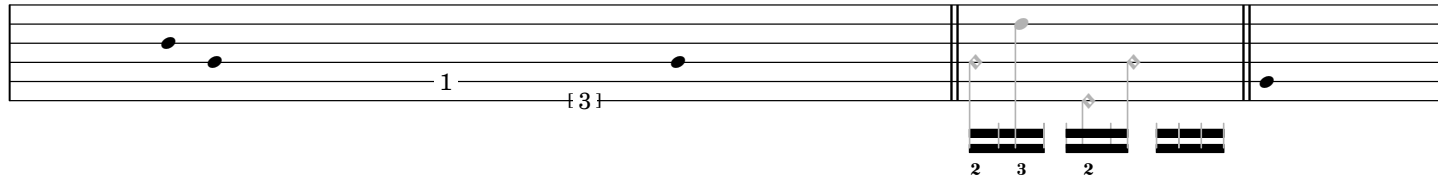
2'12"



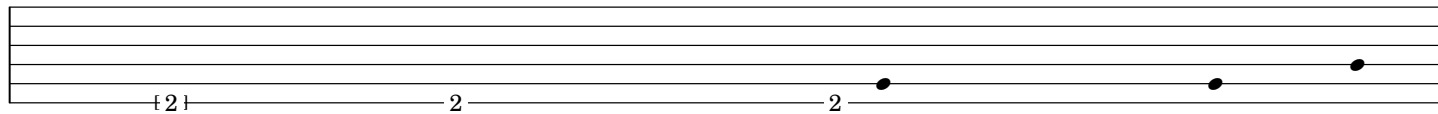
2'24"



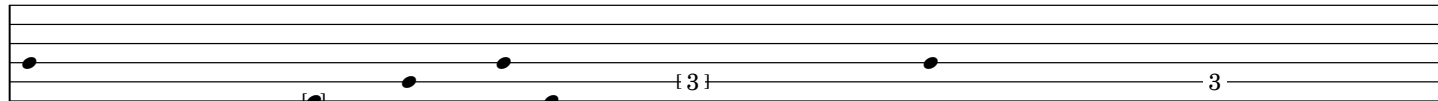
2'36"



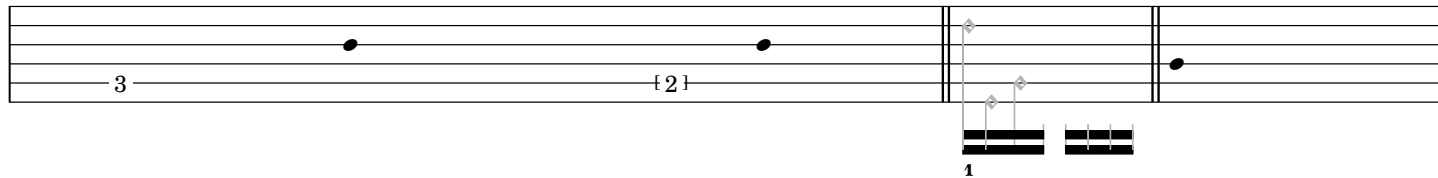
2'48"



3'00"



3'12"



A number line is shown with tick marks at -2, -1, 0, and 1. Three points are plotted and labeled as follows:

- A point at -1 is labeled $\{-1\}$.
- A point at 0 is labeled $\{-1, 0\}$.
- A point at 1 is labeled $\{-1, 0, 1\}$.

Year	Number of people (millions)
1970	3.5
1975	3.8
1980	4.2
1985	4.5
1990	4.8
1995	5.2
2000	5.5
2005	5.8
2010	6.5

The first system of the musical score for 'The Little Boat' consists of two staves. The upper staff is a treble clef with a key signature of one flat (B-flat) and a 2/4 time signature. It contains a single half note on the second line (F4). The lower staff is a bass clef with a key signature of one flat (B-flat) and a 2/4 time signature. It contains a single half note on the second space (D3). The system ends with a double bar line.

Diagram illustrating the layout of a 10-foot long instrument, divided into three sections:

- Left Section:** Labeled 5'00". It contains a brace with the number 2.
- Middle Section:** Labeled 5'04". It contains a diamond shape with a line pointing to a bracketed section of the instrument. Below this section, there are two sets of three horizontal lines, each with a vertical line through the center, and the number 1 below them.
- Right Section:** Labeled 5'00". It contains a brace with the number 2.

5'12"

A musical staff with five lines. The first line is empty. The second line has a single eighth note. The third line has a single eighth note. The fourth line has a single eighth note. The fifth line has a single eighth note. Below the staff, there are fingerings: a '2' under the first note, a '2' under the second note, and a '1' under the third note. The staff is enclosed in a box with a width of 5'12".

5'24"

1 1 { 3 } 3 3

5'48"

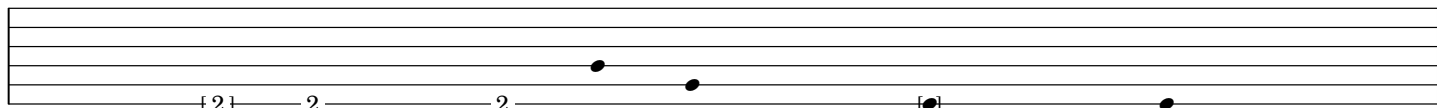
2 1 2

6'00"

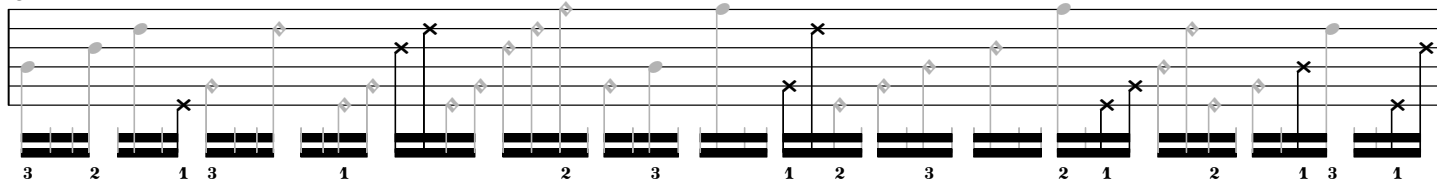
A musical staff with five lines. The first line is labeled '6'00"'. The staff contains four notes: a quarter note on the second line, a quarter note on the first space, a quarter note on the second space, and a quarter note on the third line. Below the first note is a brace with the number '3'. Below the second note is a brace with the number '3'. Below the third note is a brace with the number '2'. Below the fourth note is a brace with the number '3'.

The musical score for 'The Rose Tree' is presented on a single staff. It begins with a treble clef and a key signature of one flat (B-flat). The tempo is marked 'Allegretto' and the time signature is 6/12. The score consists of a single melodic line with various note values, including quarter, eighth, and sixteenth notes, as well as rests. The piece concludes with a double bar line and a repeat sign.

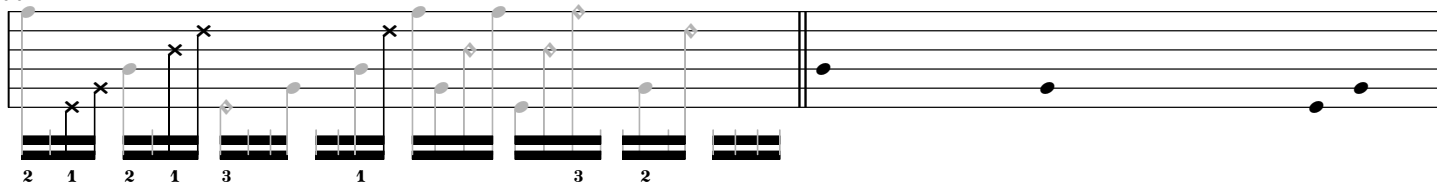
6'36"



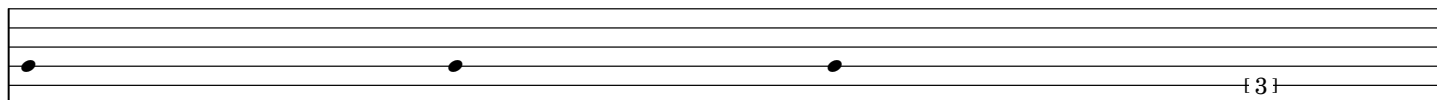
6'48"



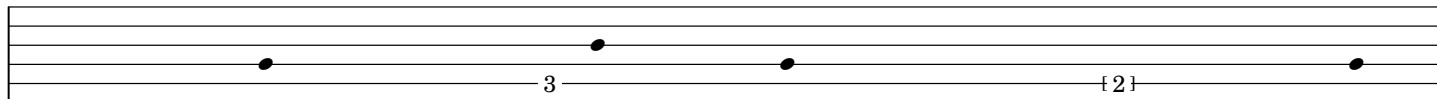
7'00"



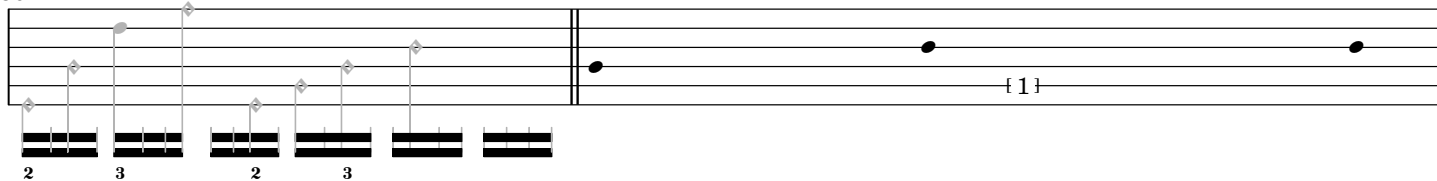
7'12"



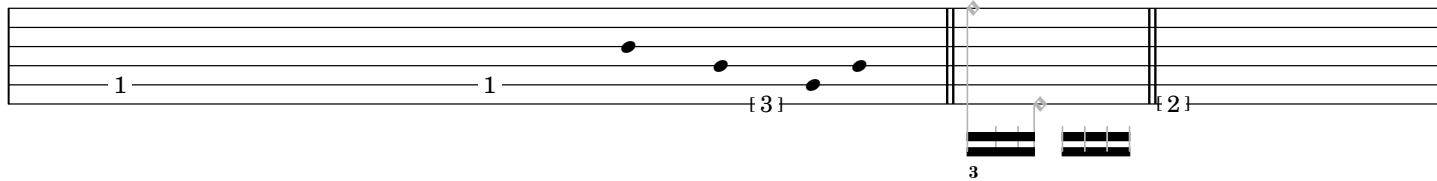
7'24"



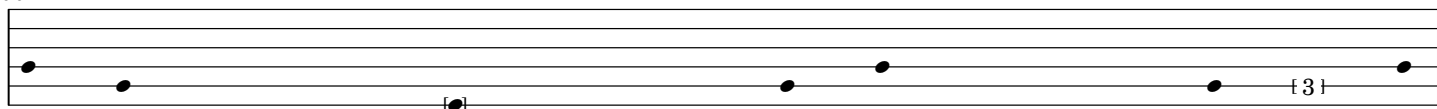
7'36"



7'48"



8'00"



8'12"

Musical notation for 8'12". The staff shows a sequence of notes with fingerings (3, 1, 3, 1, 3, 1) and a triplet of eighth notes. A double bar line is followed by a triplet of eighth notes.

8'24"

Musical notation for 8'24". The staff shows a sequence of notes with a triplet of eighth notes and a double bar line.

8'36"

8'40"

Musical notation for 8'36" and 8'40". The staff shows a sequence of notes with fingerings (1, 2, 1, 3, 1, 2, 1, 2, 1, 3) and a double bar line.

8'48"

Musical notation for 8'48". The staff shows a sequence of notes with a triplet of eighth notes and a double bar line.

9'00"

9'04"

Musical notation for 9'00" and 9'04". The staff shows a sequence of notes with fingerings (1, 3, 1, 2) and a double bar line.

9'12"

Musical notation for 9'12". The staff shows a sequence of notes.

9'24"

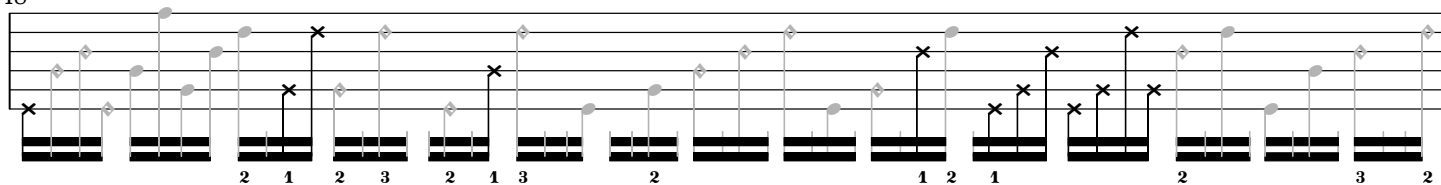
Musical notation for 9'24". The staff shows a sequence of notes.

9'36"

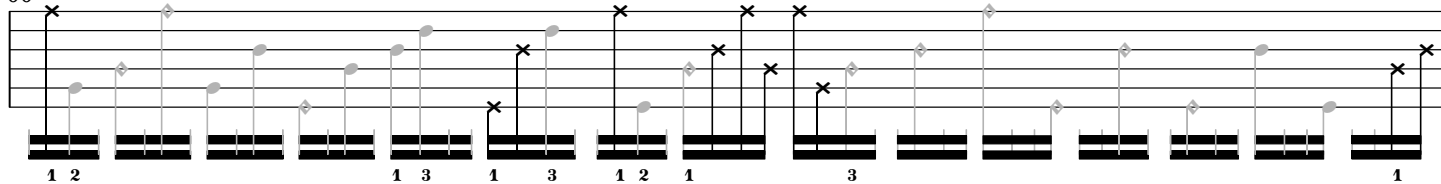
9'44"

Musical notation for 9'36" and 9'44". The staff shows a sequence of notes with fingerings (3, 1, 2, 1) and a double bar line.

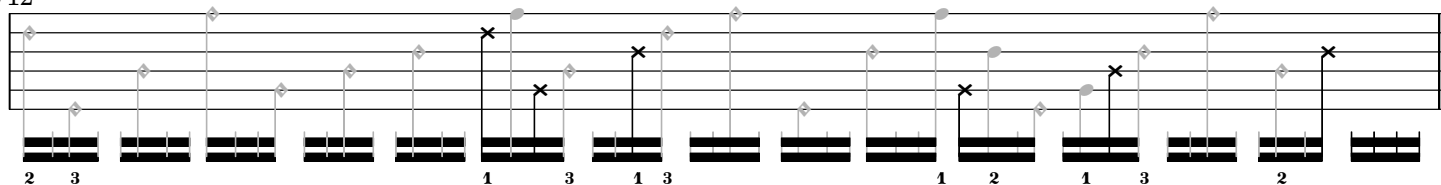
9'48"



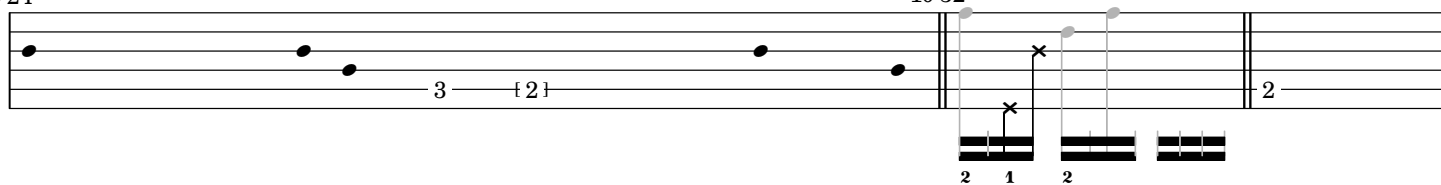
10'00"



10'12"



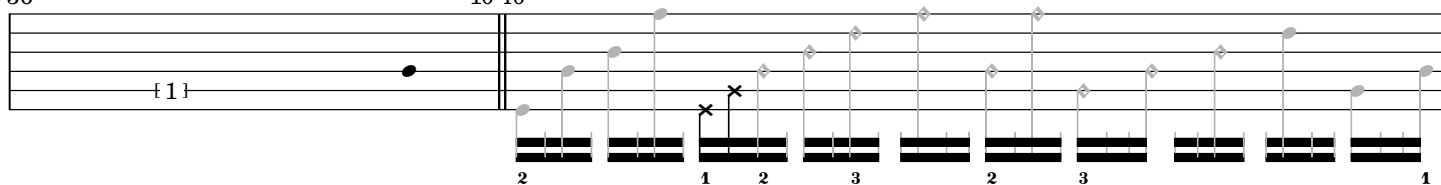
10'24"



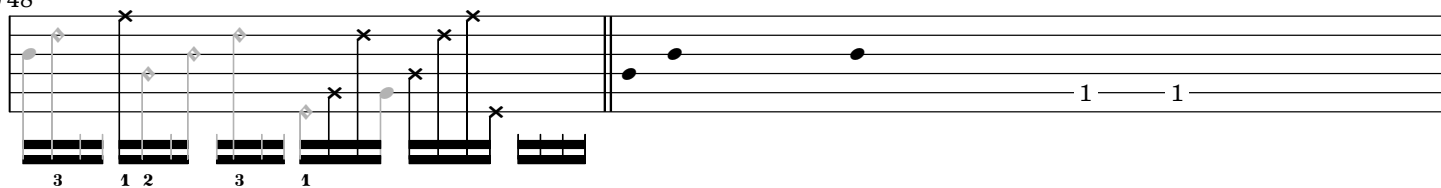
10'32"

10'36"

10'40"

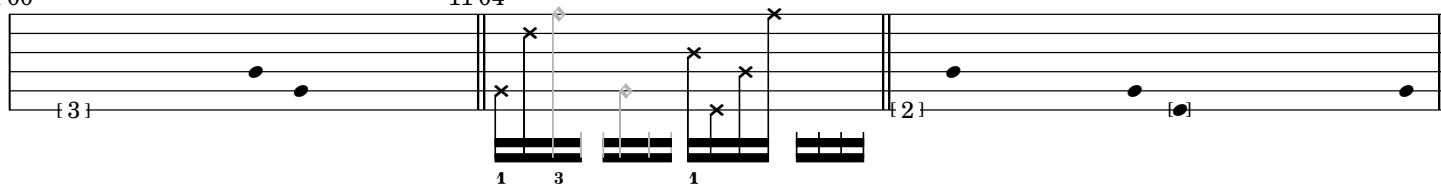


10'48"

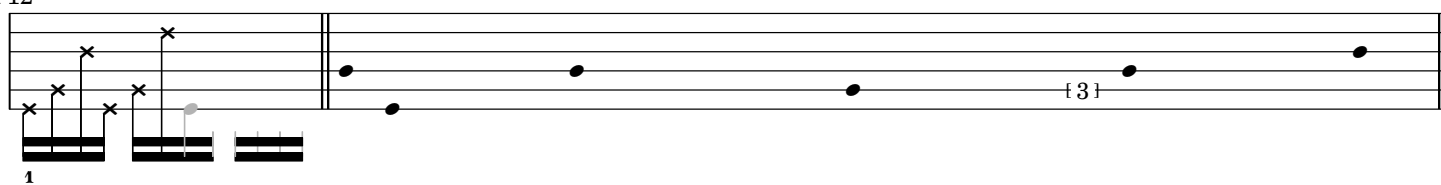


11'00"

11'04"



11'12"



11'24"

Musical notation for 11'24". The staff shows a sequence of notes with fingerings: 2, 1, 3, 2, 3, 2, 1, 3, 1, 2, 1. There are 'x' marks above some notes. The notation ends with a double bar line and a final note.

11'36"

Musical notation for 11'36". The staff shows a sequence of notes with fingerings: 1, 2, 1. There are 'x' marks above some notes. The notation ends with a double bar line and a final note.

11'44"

11'48"

Musical notation for 11'48". The staff shows a sequence of notes with fingerings: 3, {2}. There are 'x' marks above some notes. The notation ends with a double bar line and a final note.

12'00"

Musical notation for 12'00". The staff shows a sequence of notes with fingerings: 1, 2, 1, 2, 1, 2, 1, 2, 2, 1, 2, 1. There are 'x' marks above some notes. The notation ends with a double bar line and a final note.

12'12"

Musical notation for 12'12". The staff shows a sequence of notes with fingerings: {3}. There are 'x' marks above some notes. The notation ends with a double bar line and a final note.

12'24"

Musical notation for 12'24". The staff shows a sequence of notes with fingerings: 2, 1. There are 'x' marks above some notes. The notation ends with a double bar line and a final note.

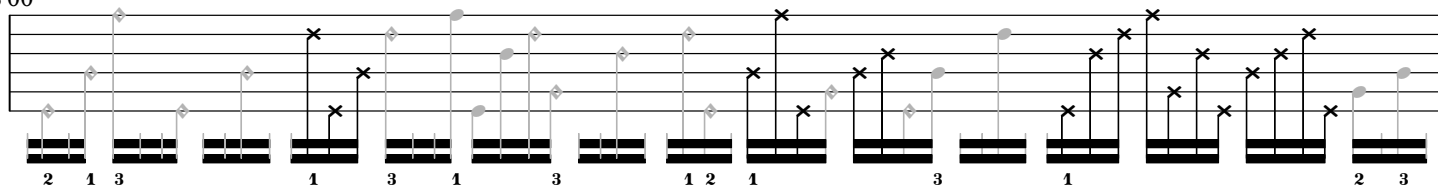
12'36"

Musical notation for 12'36". The staff shows a sequence of notes with fingerings: 3, 1, 2, {1}, 1. There are 'x' marks above some notes. The notation ends with a double bar line and a final note.

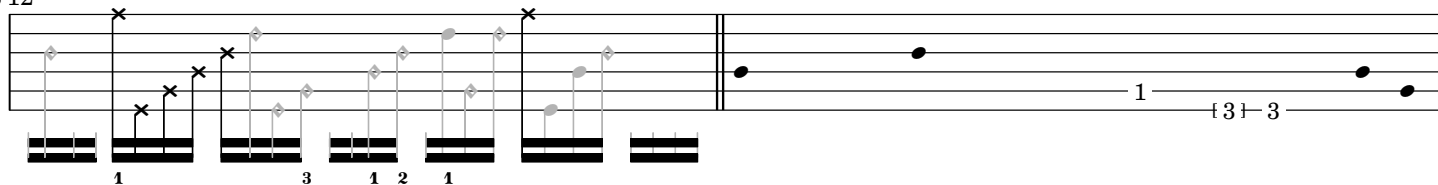
12'48"

Musical notation for 12'48". The staff shows a sequence of notes with fingerings: 2, 3, 1, 2, 1, 3, 1, 3, 1, 3, 1, 3. There are 'x' marks above some notes. The notation ends with a double bar line and a final note.

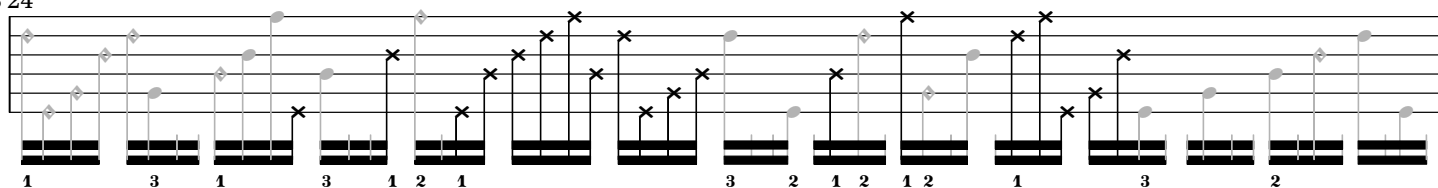
13'00"



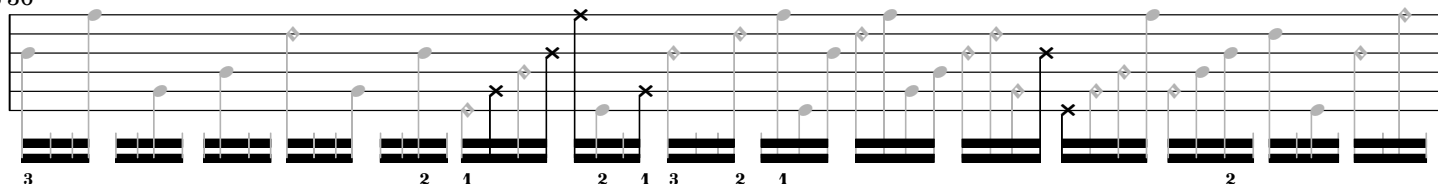
13'12"



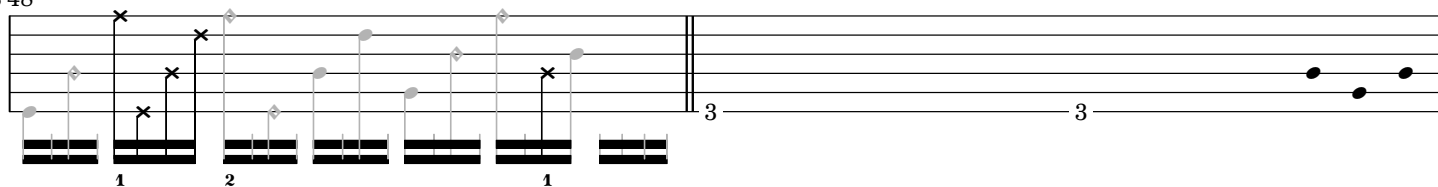
13'24"



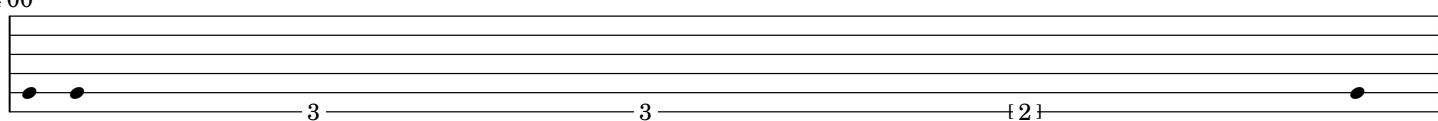
13'36"



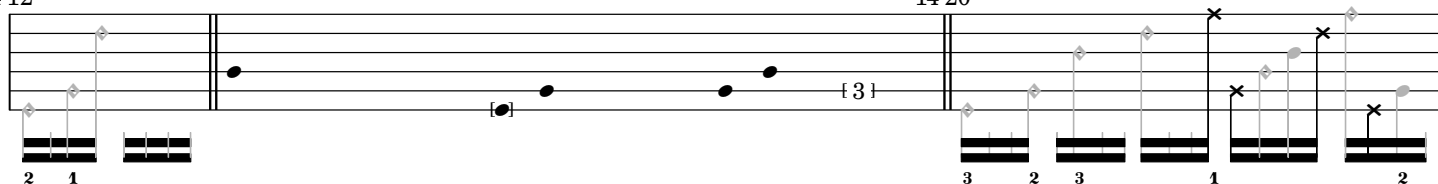
13'48"



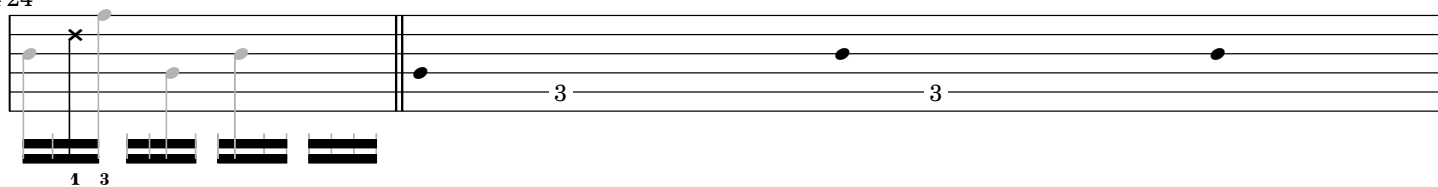
14'00"



14'12"



14'24"



14'36" 14'44"

3 { 2 }

3 1 2 3 1 2

14'48"

1 2 1 2 3 2 3 1 2 1 2 3 2 1

15'00"

3 2 1 2 1 2 3 1 3 2 1 3 2

15'12"

3 1 2 1 3 2 3 2

15'24"

2 1 2 1 2 1 3 1 2 1 1 1

15'36"

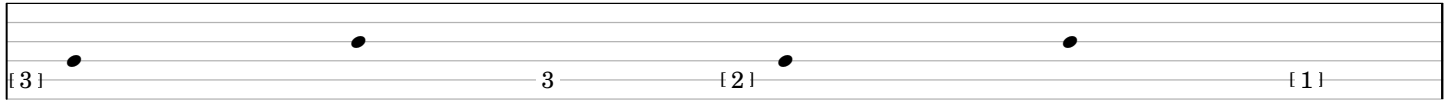
{ 2 } 2 { 3 }

ostinato and interrupt

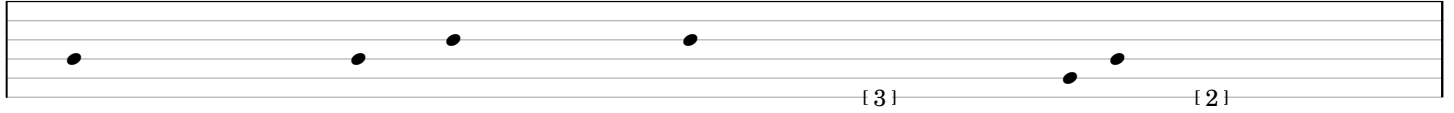
ensemble part 1

michael winter (mexico city, mx; 2017)
version generated: 2017.12.06

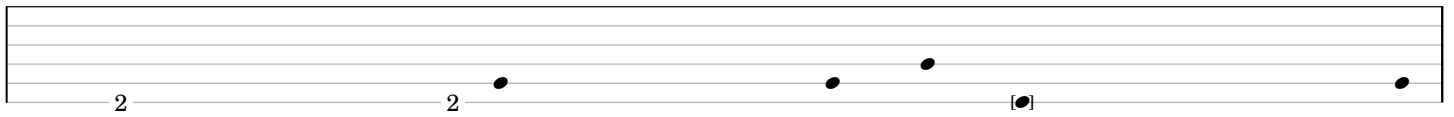
0'12"



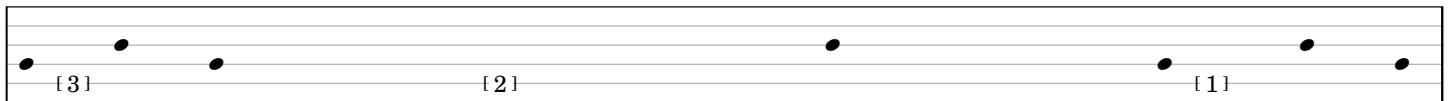
0'24"



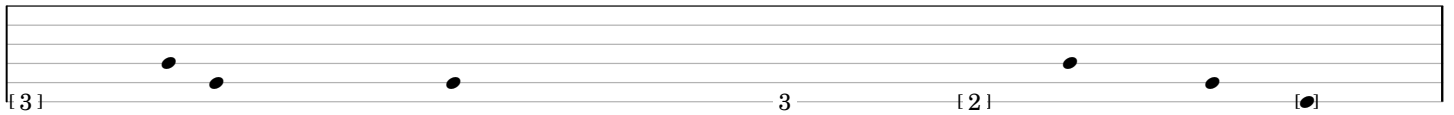
0'36"



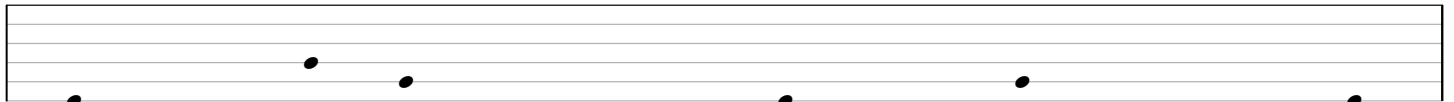
0'48"



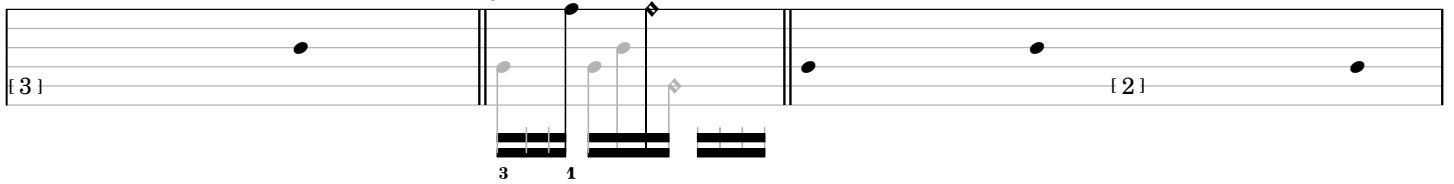
1'00"



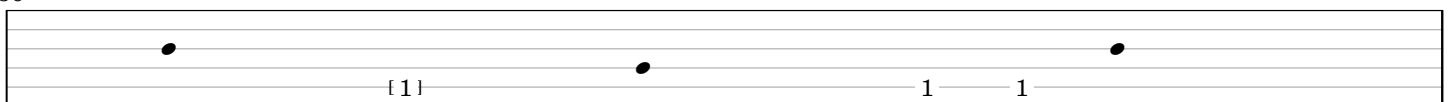
1'12"



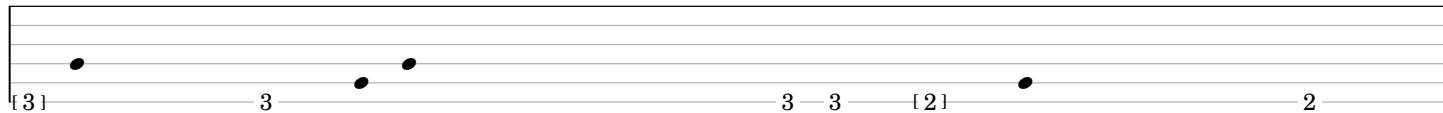
1'24"



1'36"

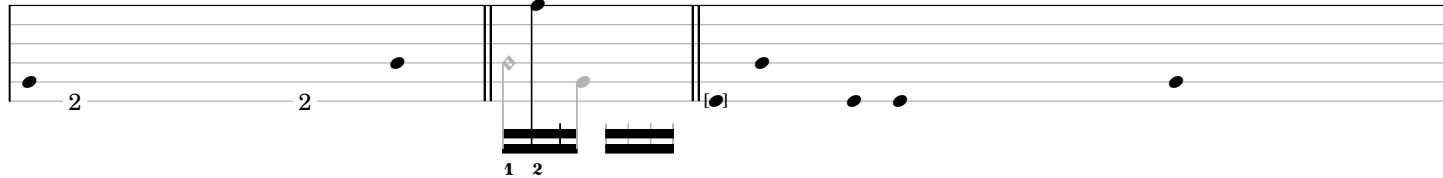


1'48"

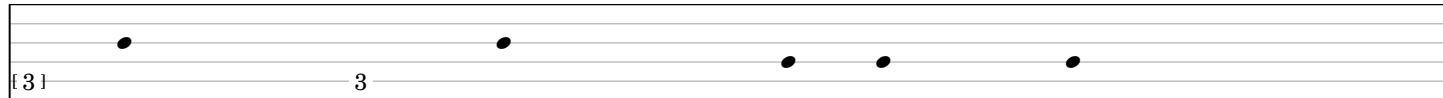


2'00"

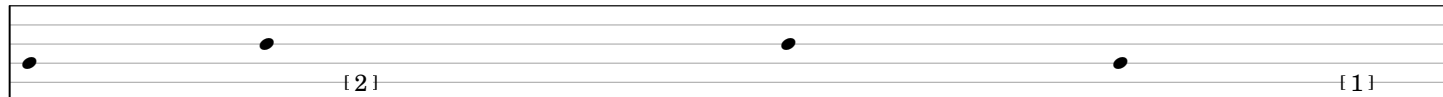
2'04"



2'12"

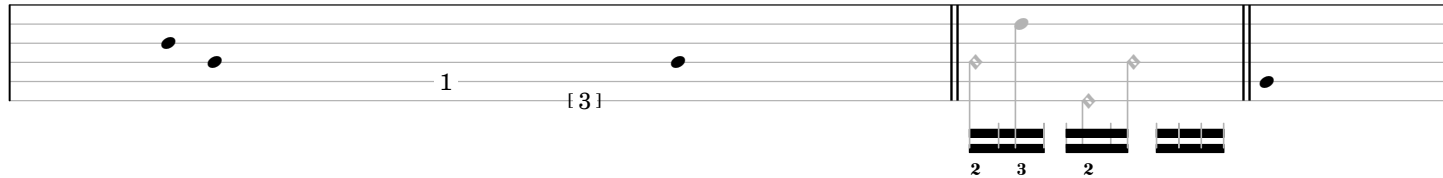


2'24"



2'36"

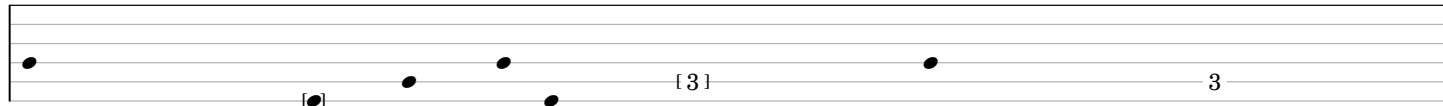
2'44"



2'48"

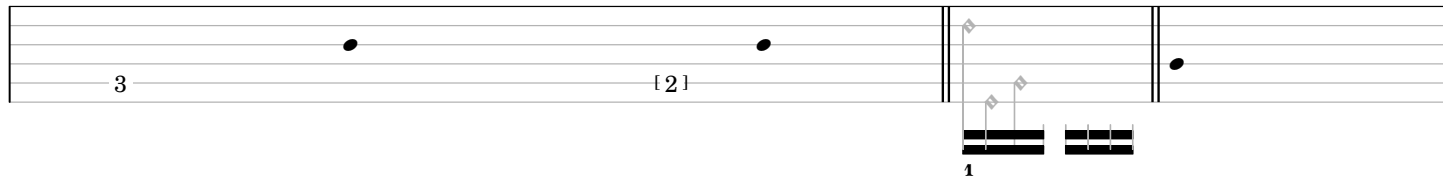


3'00"



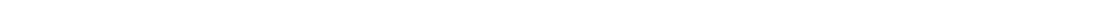
3'12"

3'20"



A number line from 0 to 10 with tick marks at every integer. Points are plotted at 2, 3, 5, 7, and 9. The points at 2 and 3 are labeled with the fraction $\frac{1}{3}$. The points at 5 and 7 are labeled with the fraction $\frac{2}{3}$. The point at 9 is labeled with the fraction $\frac{3}{3}$.

The first system of musical notation for 'The Little Boat' consists of a single staff with a treble clef and a key signature of one flat (B-flat). The melody begins with a quarter rest, followed by a quarter note G4, a quarter note A4, and a quarter note B4. This is followed by a quarter rest, then a quarter note G4, a quarter note F4, and a quarter note E4. The system concludes with a quarter note D4, a quarter note C4, and a quarter note B3, with a final double bar line.



A number line from 0 to 10. Two points are plotted: one at 1 and one at 3. The point at 1 is labeled '1' below it. The point at 3 is labeled '[3] 3' below it.

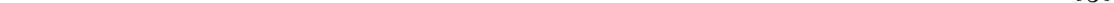
The first system of musical notation for 'The Little Boat' consists of a single staff. It begins with a treble clef and a key signature of one flat (B-flat). The melody starts with a quarter note G4, followed by a quarter note A4, and then a quarter note B4. This is followed by a half note G4, and then a half note F4. The melody concludes with a quarter note E4. The system is marked with a '3' below the first measure and a '[2]' below the fifth measure, indicating a 3-measure rest and a 2-measure rest, respectively.

The first system of musical notation for 'The Little Boat' consists of a treble clef, a key signature of one flat (B-flat), and a 2/4 time signature. The melody is written on a five-line staff. It begins with a quarter note G4, followed by a quarter note A4, then a quarter note B-flat4. The next measure contains a quarter note C5, a quarter note B-flat4, and a quarter note A4. The final measure of the system contains a quarter note G4, a quarter note F4, and a quarter note E4. The system ends with a double bar line.

A diagram showing a sequence of points on a horizontal line. From left to right: a point, a point labeled $[3]$, a point labeled 3 , a point, a point labeled 3 , a point labeled $[2]$, and a point.

The musical score for 'The Rose Tree' is presented on a single staff. It begins with a treble clef and a key signature of one flat (B-flat). The melody is written in a simple, folk-like style. The first line of the melody is: G4 (quarter), A4 (quarter), B4 (quarter), A4-G4 (beamed eighth notes), F4 (quarter), E4 (quarter), D4 (half). The second line is: C4 (half), B3 (quarter), A3 (quarter), G3 (quarter), F3 (quarter), E3 (quarter), D3 (half). The third line is: C3 (half), B2 (quarter), A2 (quarter), G2 (quarter), F2 (quarter), E2 (quarter), D2 (half). The fourth line is: C2 (half), B1 (quarter), A1 (quarter), G1 (quarter), F1 (quarter), E1 (quarter), D1 (half). The fifth line is: C1 (half), B0 (quarter), A0 (quarter), G0 (quarter), F0 (quarter), E0 (quarter), D0 (half). The sixth line is: C0 (half), B-1 (quarter), A-1 (quarter), G-1 (quarter), F-1 (quarter), E-1 (quarter), D-1 (half). The seventh line is: C-1 (half), B-2 (quarter), A-2 (quarter), G-2 (quarter), F-2 (quarter), E-2 (quarter), D-2 (half). The eighth line is: C-2 (half), B-3 (quarter), A-3 (quarter), G-3 (quarter), F-3 (quarter), E-3 (quarter), D-3 (half). The ninth line is: C-3 (half), B-4 (quarter), A-4 (quarter), G-4 (quarter), F-4 (quarter), E-4 (quarter), D-4 (half). The tenth line is: C-4 (half), B-5 (quarter), A-5 (quarter), G-5 (quarter), F-5 (quarter), E-5 (quarter), D-5 (half). The score concludes with a double bar line.

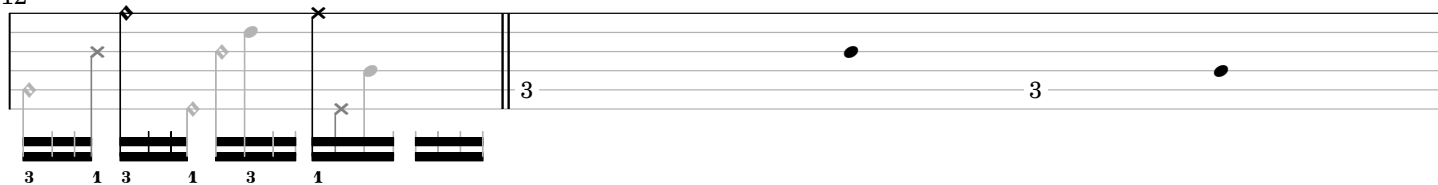
The figure shows a plot of the function $f_3(x) = 1 - 3x + 3x^2 - x^3$. The x-axis is labeled with $[3]$ and 3 . The y-axis has a tick mark at 1 . The curve starts at $(0, 1)$, decreases to a minimum at $x=1$, and then increases to a maximum at $x=3$.



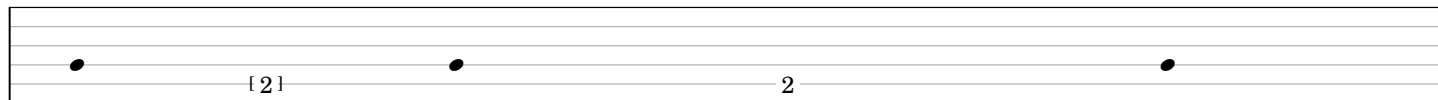
The first system of the musical score for 'The Little Boat' consists of two staves. The upper staff is a treble clef with a key signature of one flat (B-flat). It contains a melody starting on a whole note G4, followed by a half note F4, and then a quarter note E4. The lower staff is a bass clef with a key signature of one flat (B-flat). It contains a bass line starting on a whole note D3, followed by a half note C3, and then a quarter note B2. The system ends with a double bar line.

The diagram illustrates a sequence of points on a horizontal line. The points are labeled 1, 1, [3], and [2]. Below the line, there are two sets of three horizontal bars, each labeled 3. A vertical line separates the points from the bars.

8'12"

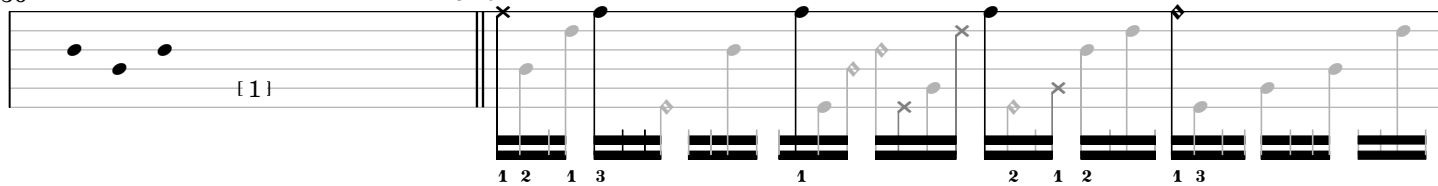


8'24"

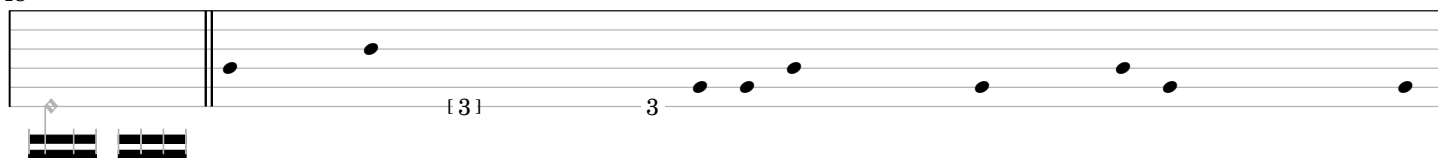


8'36"

8'40"

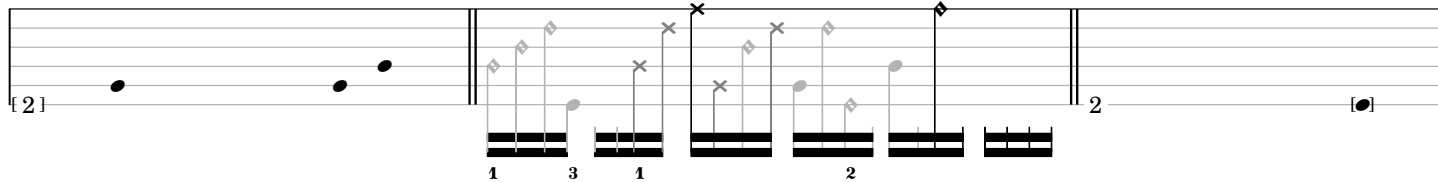


8'48"



9'00"

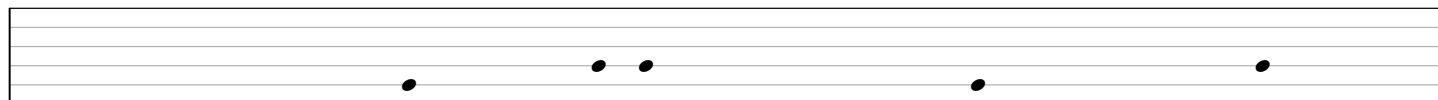
9'04"



9'12"

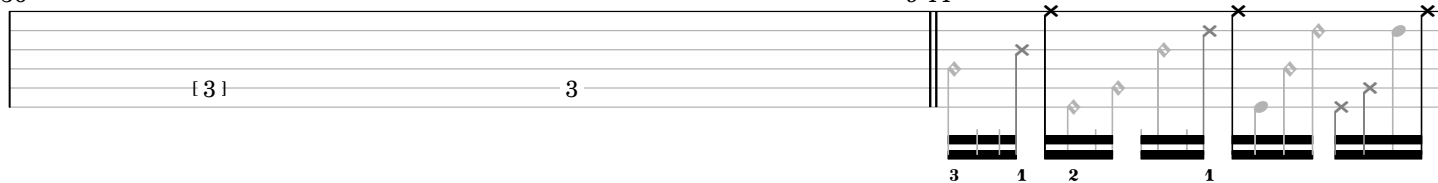


9'24"

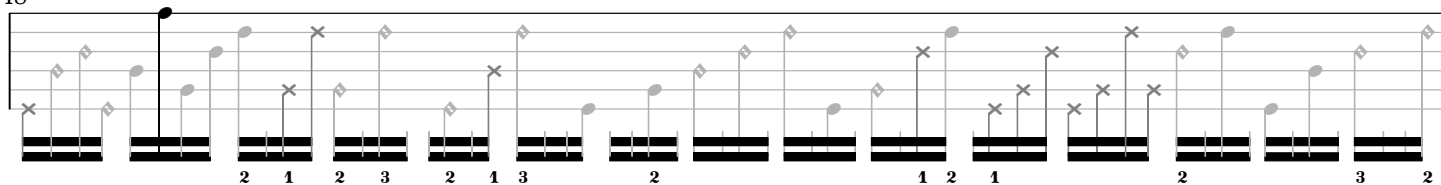


9'36"

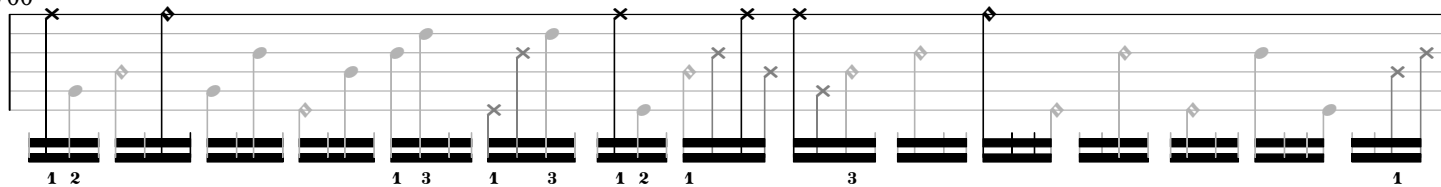
9'44"



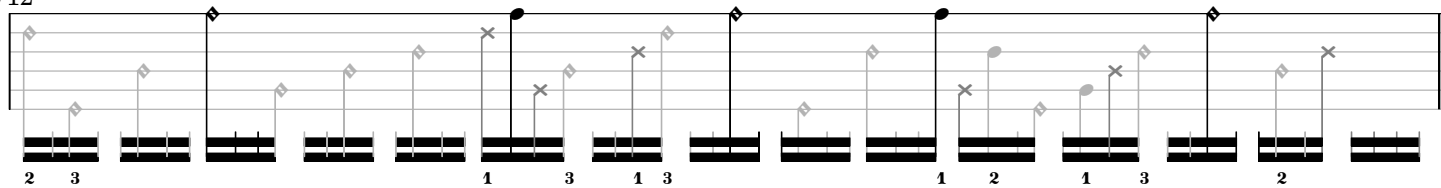
9'48"



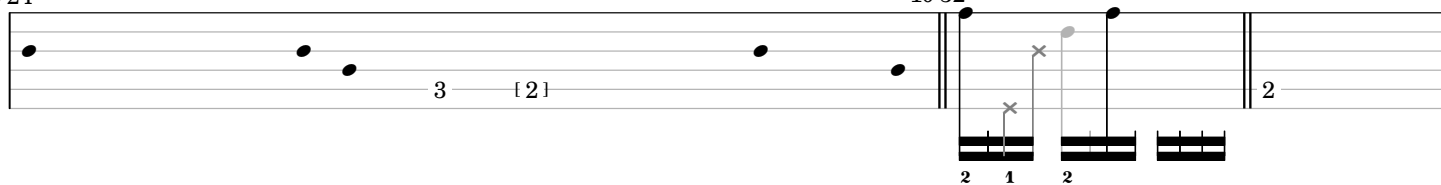
10'00"



10'12"

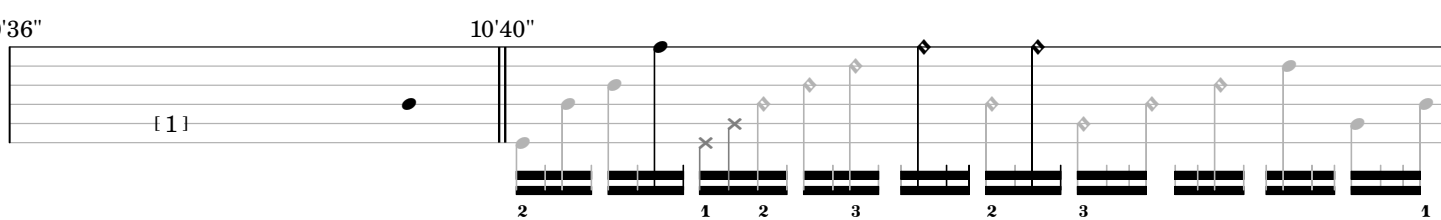


10'24"



10'32"

10'36"

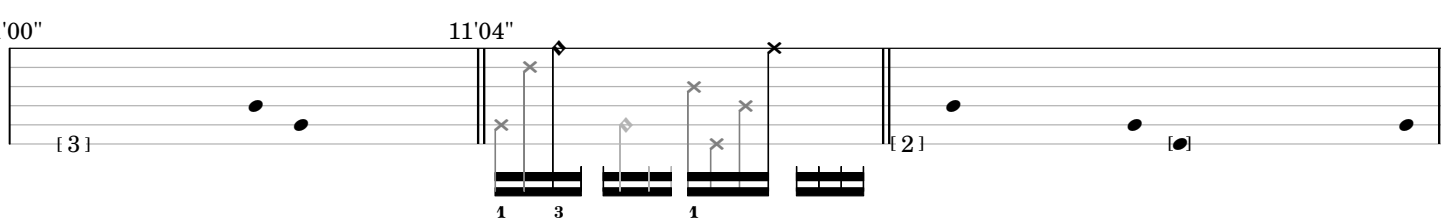


10'40"

10'48"

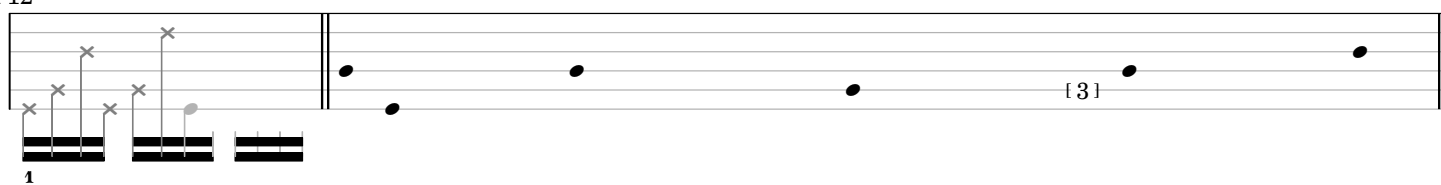


11'00"



11'04"

11'12"



11'24"

Musical notation for 11'24". The staff shows a sequence of notes with various fingerings (2, 1, 3, 2, 3, 2, 1, 3, 1, 2, 1) and articulation marks (x, diamond). The notation is split into two measures by a double bar line. The first measure contains a sequence of notes with fingerings 2, 1, 3, 2, 3, 2, 1, 3, 1, 2, 1. The second measure contains a sequence of notes with fingerings [2] and [1].

11'36"

Musical notation for 11'36". The staff shows a sequence of notes with various fingerings (1, 2, 1) and articulation marks (x, diamond). The notation is split into two measures by a double bar line. The first measure contains a sequence of notes with fingerings 1, 2, 1. The second measure contains a sequence of notes with fingerings [3].

11'44"

11'48"

Musical notation for 11'48". The staff shows a sequence of notes with various fingerings (3, [2]) and articulation marks (x, diamond). The notation is split into two measures by a double bar line. The first measure contains a sequence of notes with fingerings 3, [2]. The second measure contains a sequence of notes with fingerings [2].

12'00"

Musical notation for 12'00". The staff shows a sequence of notes with various fingerings (1, 2, 1, 2, 1, 2, 1, 2, 2, 1, 2, 1) and articulation marks (x, diamond). The notation is split into two measures by a double bar line. The first measure contains a sequence of notes with fingerings 1, 2, 1, 2, 1, 2, 1, 2, 2, 1, 2, 1. The second measure contains a sequence of notes with fingerings [2].

12'12"

Musical notation for 12'12". The staff shows a sequence of notes with various fingerings (3) and articulation marks (x, diamond). The notation is split into two measures by a double bar line. The first measure contains a sequence of notes with fingerings 3. The second measure contains a sequence of notes with fingerings [3].

12'24"

Musical notation for 12'24". The staff shows a sequence of notes with various fingerings (2, 1) and articulation marks (x, diamond). The notation is split into two measures by a double bar line. The first measure contains a sequence of notes with fingerings 2, 1. The second measure contains a sequence of notes with fingerings [2].

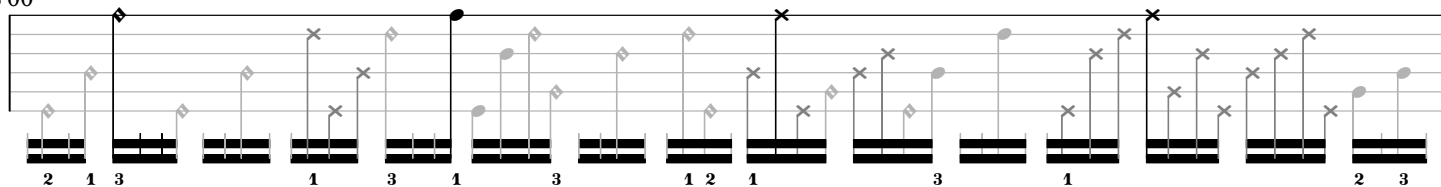
12'36"

Musical notation for 12'36". The staff shows a sequence of notes with various fingerings (3, 1, 2) and articulation marks (x, diamond). The notation is split into two measures by a double bar line. The first measure contains a sequence of notes with fingerings 3, 1, 2. The second measure contains a sequence of notes with fingerings [1], 1.

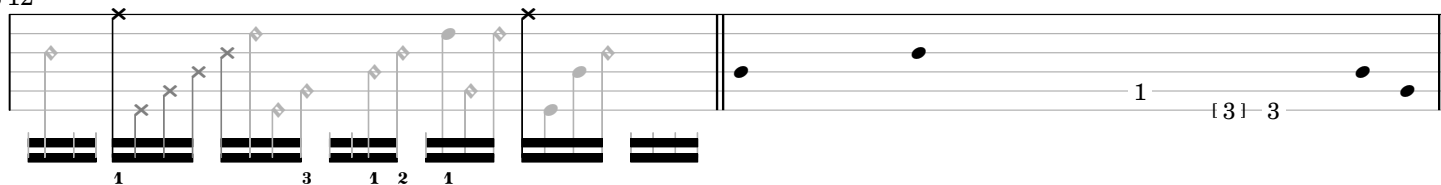
12'48"

Musical notation for 12'48". The staff shows a sequence of notes with various fingerings (2, 3, 1, 2, 1, 3, 1, 3, 1, 3) and articulation marks (x, diamond). The notation is split into two measures by a double bar line. The first measure contains a sequence of notes with fingerings 2, 3, 1, 2, 1, 3, 1, 3. The second measure contains a sequence of notes with fingerings 1, 3.

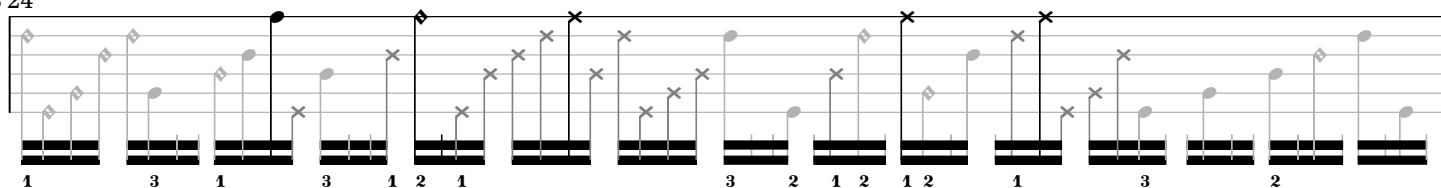
13'00"



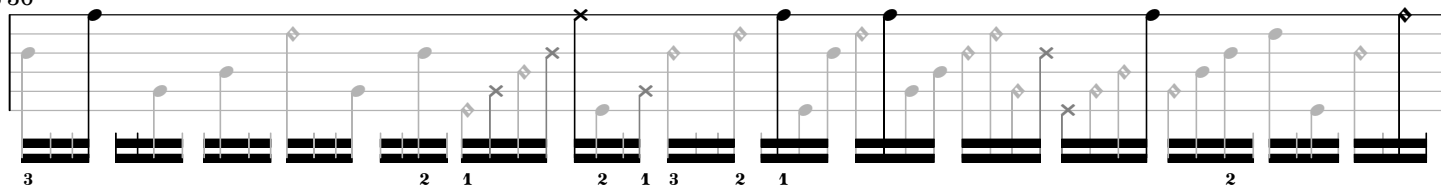
13'12"



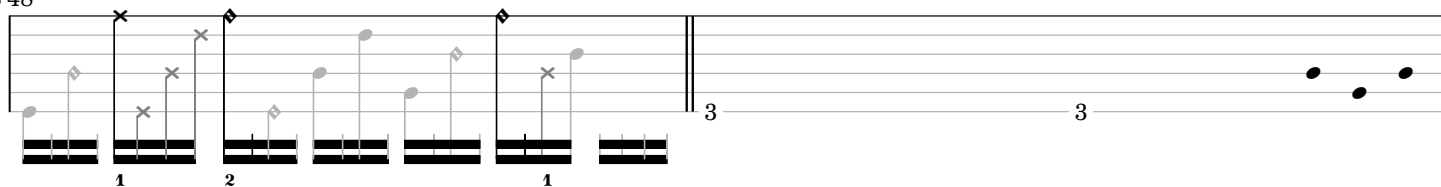
13'24"



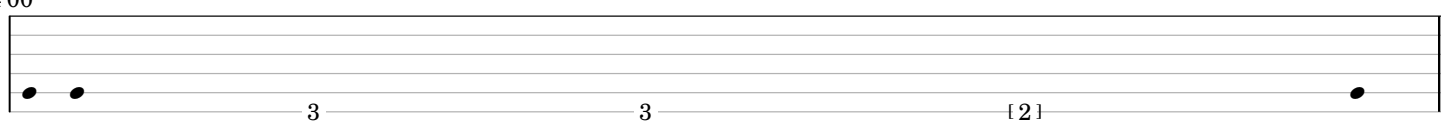
13'36"



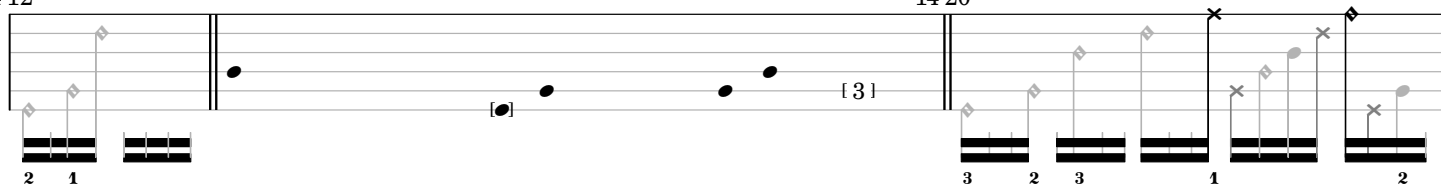
13'48"



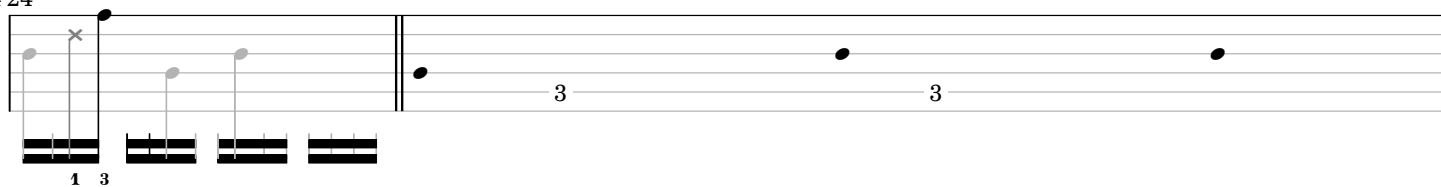
14'00"



14'12"



14'24"



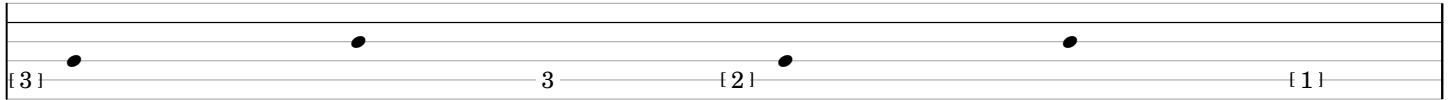
[illegible]

ostinato and interrupt

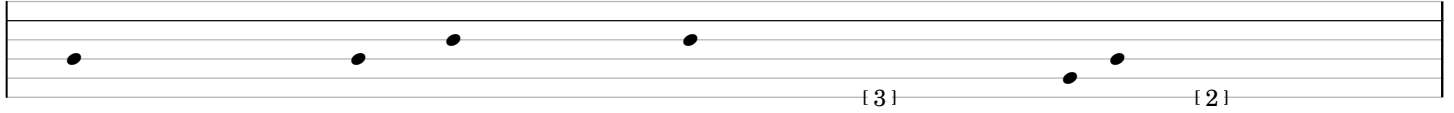
ensemble part 2

michael winter (mexico city, mx; 2017)
version generated: 2017.12.06

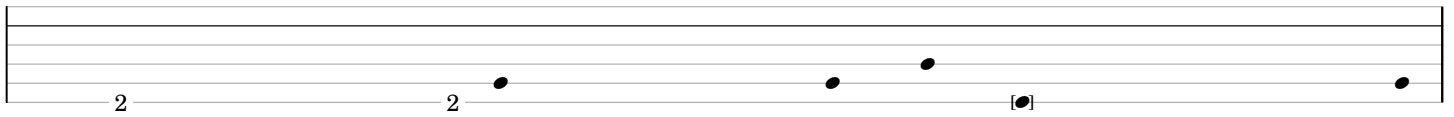
0'12"



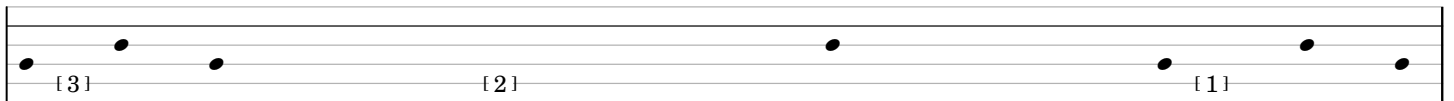
0'24"



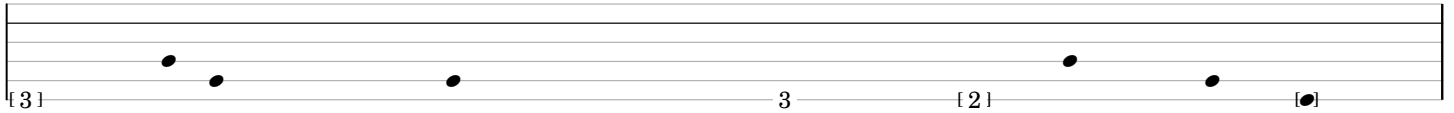
0'36"



0'48"



1'00"

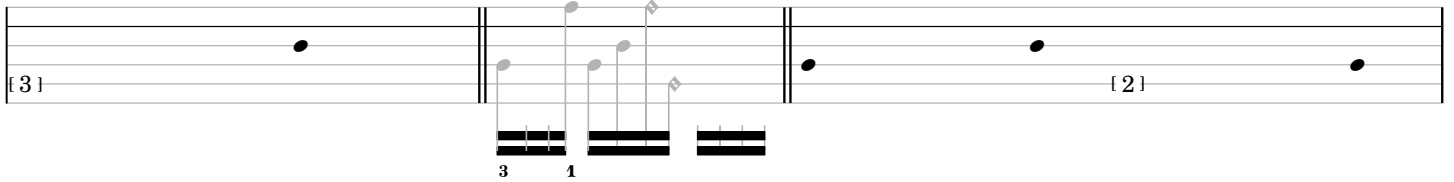


1'12"

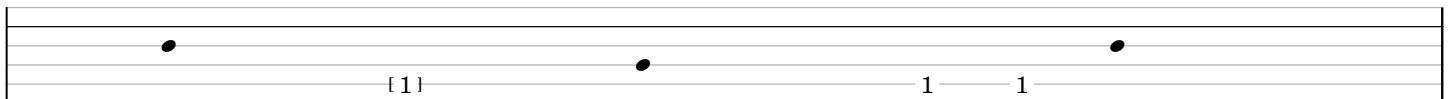


1'24"

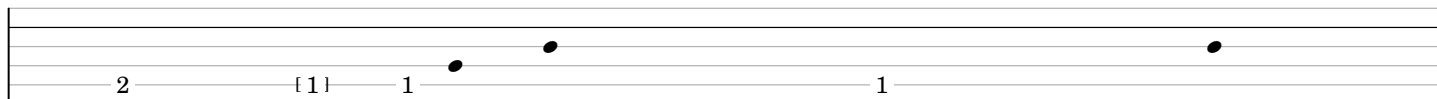
1'28"



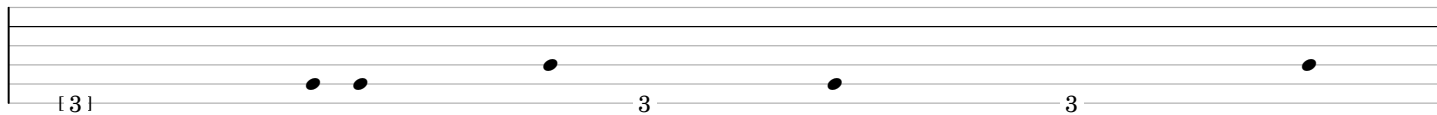
1'36"



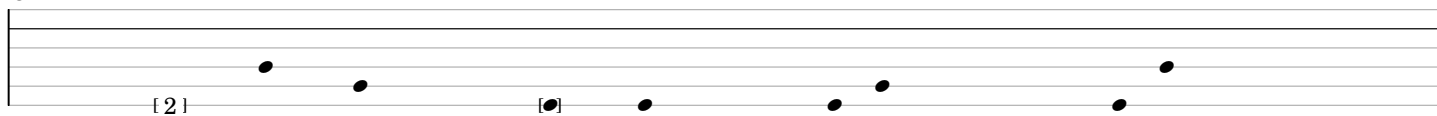
3'24"



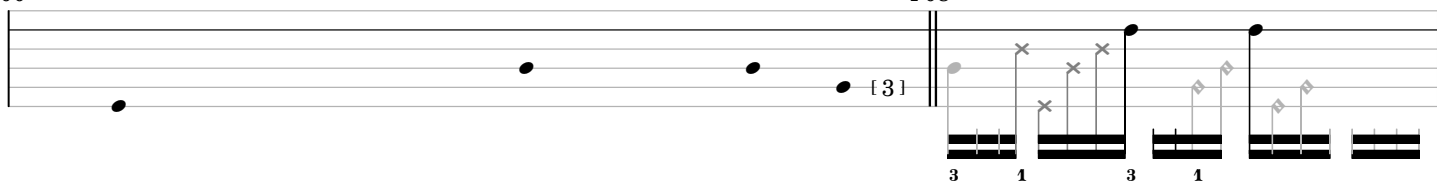
3'36"



3'48"

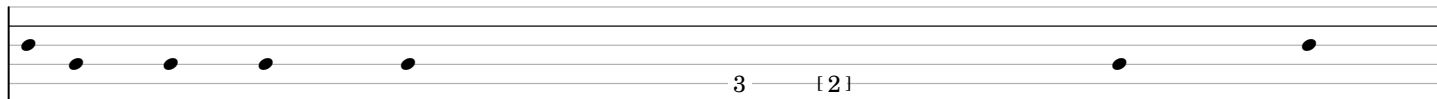


4'00"

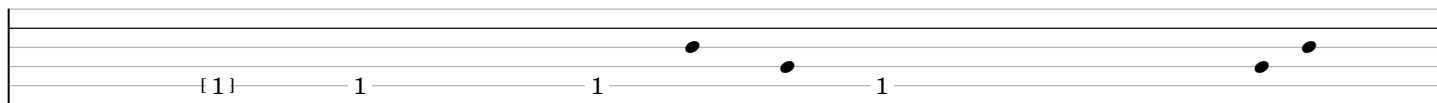


4'08"

4'12"



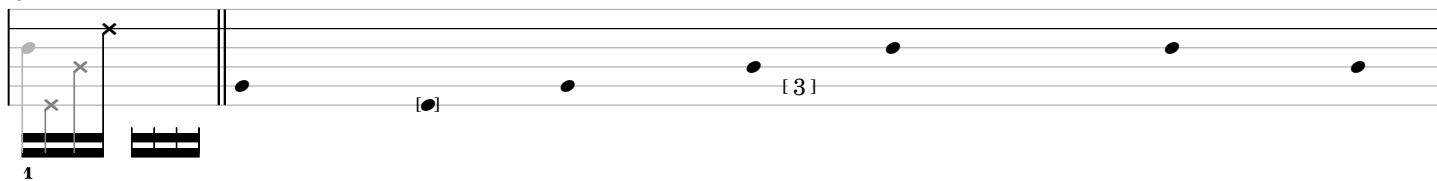
4'24"



4'36"



4'48"



5'00" 5'04"

5'12"

5'24"

5'36" 5'44"

5'48"

6'00"

6'12" 6'16"

6'24"

The musical notation for the 'Trio' section is presented on a single staff. It begins with a treble clef and a key signature of one flat (B-flat). The tempo is marked 'Allegretto' and the time signature is 3/4. The notation includes various note values (quarter, eighth, and sixteenth notes), rests, and dynamic markings such as 'p' (piano) and 'f' (forte). The section concludes with a double bar line.

The first system of the musical score for 'The Little Boat' consists of two staves. The upper staff is a treble clef with a key signature of one flat (B-flat) and a 2/4 time signature. It contains a melody starting on a whole note G4, followed by a half note F4, and then a half note E4. The lower staff is a bass clef with a key signature of one flat (B-flat) and a 2/4 time signature. It contains a bass line starting on a whole note G3, followed by a half note F3, and then a half note E3. The system ends with a double bar line.

The musical score for 'The Rose Tree' is presented on a single staff. The melody begins with a quarter rest, followed by a quarter note G4, a quarter note A4, a quarter note B4, and a quarter note A4. This is followed by a quarter note G4, a quarter note F#4, a quarter note E4, and a quarter note D4. The melody then continues with a quarter note C4, a quarter note B3, a quarter note A3, and a quarter note G3. The piece concludes with a quarter note F#3, a quarter note E3, and a quarter note D3. The key signature is one sharp (F#), and the time signature is 4/4. The score is written in a simple, clear font, with the notes and rests clearly marked on the staff lines.

[illegible]

8'12"

Musical notation for 8'12". The staff shows a sequence of notes with various articulations (diamonds, crosses) and fingerings (3, 1, 3, 1, 3, 1). A double bar line is present. The notation includes a '3' below the staff and a '3' above the staff.

8'24"

Musical notation for 8'24". The staff shows a sequence of notes with fingerings (2, 2) and a double bar line.

8'36"

8'40"

Musical notation for 8'36" and 8'40". The staff shows a sequence of notes with various articulations (diamonds, crosses) and fingerings (1, 2, 1, 3, 1, 2, 1, 2, 1, 3). A double bar line is present.

8'48"

Musical notation for 8'48". The staff shows a sequence of notes with fingerings (3, 3) and a double bar line.

9'00"

9'04"

Musical notation for 9'00" and 9'04". The staff shows a sequence of notes with various articulations (diamonds, crosses) and fingerings (2, 1, 3, 1, 2). A double bar line is present.

9'12"

Musical notation for 9'12". The staff shows a sequence of notes.

9'24"

Musical notation for 9'24". The staff shows a sequence of notes.

9'36"

9'44"

Musical notation for 9'36" and 9'44". The staff shows a sequence of notes with various articulations (diamonds, crosses) and fingerings (3, 1, 2, 1). A double bar line is present.

The musical notation for the 'Trio' section is presented on a single staff. It begins with a treble clef and a key signature of one flat (B-flat). The tempo is marked 'Allegretto' and the time signature is 3/4. The notation includes various note values (quarter, eighth, and sixteenth notes), rests, and dynamic markings such as 'p' (piano) and 'f' (forte). The piece concludes with a double bar line.

The first system of the musical score for 'The Little Boat' consists of a single staff. It begins with a treble clef, a key signature of one flat (B-flat), and a common time signature (C). The melody is written in a simple, child-friendly style. The first measure contains a quarter note on G4. The second measure contains a quarter note on A4. The third measure contains a quarter note on B4. The fourth measure contains a quarter note on A4. The fifth measure contains a quarter note on G4. The sixth measure contains a quarter note on F4. The seventh measure contains a quarter note on E4. The eighth measure contains a quarter note on D4. The ninth measure contains a quarter note on C4. The tenth measure contains a quarter note on B3. The eleventh measure contains a quarter note on A3. The twelfth measure contains a quarter note on G3. The thirteenth measure contains a quarter note on F3. The fourteenth measure contains a quarter note on E3. The fifteenth measure contains a quarter note on D3. The sixteenth measure contains a quarter note on C3. The system ends with a double bar line.

The first system of the musical score for 'The Rose Tree' consists of a single staff. It begins with a treble clef and a key signature of one flat (B-flat). The melody starts with a half rest, followed by a quarter note G4, a quarter note A4, and a quarter note B4. The lyrics 'The Rose Tree' are written below the staff, aligned with the notes. The system ends with a double bar line.

11'24"

The first system of musical notation for 'The Little Boat' consists of a single staff with a treble clef and a key signature of one flat (B-flat). The melody is written in a simple, child-friendly style. It begins with a quarter note on G4, followed by a quarter note on A4, and then a quarter note on B4. This is followed by a quarter note on A4, a quarter note on G4, and a quarter note on F4. The melody continues with a quarter note on E4, a quarter note on D4, and a quarter note on C4. The system ends with a double bar line. Below the staff, there are fingerings: '2' under the first G, '1' under the first A, '3' under the first B, '2' under the second A, '3' under the second G, and '1' under the first C.

11'36"

The first system of musical notation for 'The Rose Tree' consists of two staves. The top staff contains a melody with notes marked with 'x' and 'o' symbols. The bottom staff contains a bass line with notes marked with 'x' and 'o' symbols. The system is divided into two measures by a double bar line. The first measure contains a melody starting on a middle C, moving up to a D, then a B, and finally a C. The second measure contains a melody starting on a middle C, moving up to a D, then a B, and finally a C. The bass line in the first measure starts on a G, moves up to an A, then a B, and finally a C. The bass line in the second measure starts on a G, moves up to an A, then a B, and finally a C. The system is labeled with '10' and '11 44' at the top.

11'44"

11'48"

Transportation Method	Number of People
Bicycles	2
Buses	10
Cars	12
Motorcycles	14
Trains	1

12'00"

12'12"

12'24"

The first system of the musical score for 'The Little Boat' consists of a single staff with a treble clef and a key signature of one flat (B-flat). The melody begins with a quarter note on G4, followed by a quarter note on A4, and then a quarter note on B4. This is followed by a quarter note on A4, a quarter note on G4, and a quarter note on F4. The melody then moves to a half note on E4. The first system ends with a double bar line. Below the staff, there are two sets of fingerings: '2' under the first G4 and '1' under the first A4. There are also two sets of fingerings: '2' under the first G4 and '1' under the first A4. There are also two sets of fingerings: '2' under the first G4 and '1' under the first A4.

12'36"

12'48"

13'00"

Musical notation for 13'00. The staff shows a sequence of notes with various articulations (diamonds, crosses, dots) and fingerings (2, 1, 3, 1, 3, 1, 3, 1, 2, 1, 3, 1, 3, 2, 3). The notes are mostly eighth and sixteenth notes, with some quarter notes.

13'12"

Musical notation for 13'12. The staff shows a sequence of notes with various articulations (diamonds, crosses, dots) and fingerings (1, 3, 1, 2, 1). The notation includes a double bar line and a measure with a whole note and a triplet of eighth notes (1 [3] 3).

13'24"

Musical notation for 13'24. The staff shows a sequence of notes with various articulations (diamonds, crosses, dots) and fingerings (1, 3, 1, 3, 1, 2, 1, 3, 2, 1, 2, 1, 2, 1, 3, 2). The notes are mostly eighth and sixteenth notes, with some quarter notes.

13'36"

Musical notation for 13'36. The staff shows a sequence of notes with various articulations (diamonds, crosses, dots) and fingerings (3, 2, 1, 2, 2, 1, 3, 2, 1, 2, 2). The notes are mostly eighth and sixteenth notes, with some quarter notes.

13'48"

Musical notation for 13'48. The staff shows a sequence of notes with various articulations (diamonds, crosses, dots) and fingerings (1, 2, 1). The notation includes a double bar line and a measure with a whole note and a triplet of eighth notes (3 3).

14'00"

Musical notation for 14'00. The staff shows a sequence of notes with various articulations (diamonds, crosses, dots) and fingerings (3, 3, [2]). The notation includes a double bar line and a measure with a whole note and a triplet of eighth notes (3 3).

14'12"

Musical notation for 14'12. The staff shows a sequence of notes with various articulations (diamonds, crosses, dots) and fingerings (2, 1, 3, 2, 3, 1, 2). The notation includes a double bar line and a measure with a whole note and a triplet of eighth notes (3 3).

14'20"

14'24"

Musical notation for 14'24. The staff shows a sequence of notes with various articulations (diamonds, crosses, dots) and fingerings (1, 3). The notation includes a double bar line and a measure with a whole note and a triplet of eighth notes (3 3).

[illegible]

The figure shows a musical score for two staves. The top staff has a treble clef and contains several notes with stems and flags. Some notes are marked with an 'x' or a 'y'. The bottom staff has a bass clef and contains rhythmic notation represented by vertical lines. Below the bottom staff, there are numbers: 3, 1, 2, 3, 1, 2, which likely correspond to the rhythmic values of the notes above.

The second staff contains a sequence of notes and rests. The notes are: a quarter note (fingering 3), an eighth note (fingering 1), a quarter note (fingering 2), an eighth note (fingering 1), a quarter note (fingering 3), an eighth note (fingering 2), a quarter note (fingering 3), an eighth note (fingering 2), and a final measure with a whole rest and a '1' below it.

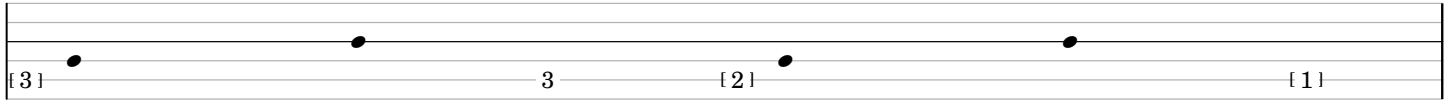
A diagram of a 2D coordinate system with a horizontal axis and a vertical axis. The horizontal axis has a point labeled $[2]$ and a point labeled 2 . The vertical axis has a point labeled $[0]$. There are five black dots: one at $[2]$, one at 2 , one at $[0]$, and two others in the first quadrant.

ostinato and interrupt

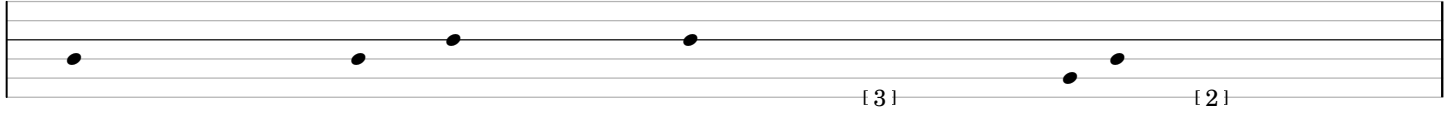
ensemble part 3

michael winter (mexico city, mx; 2017)
version generated: 2017.12.06

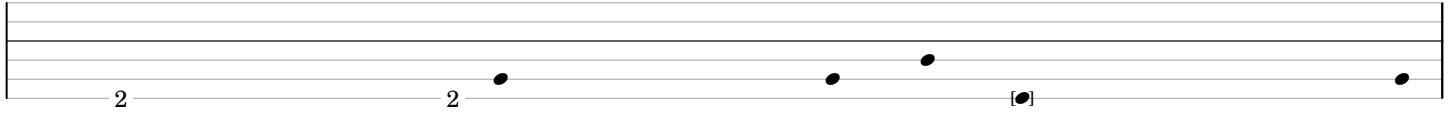
0'12"



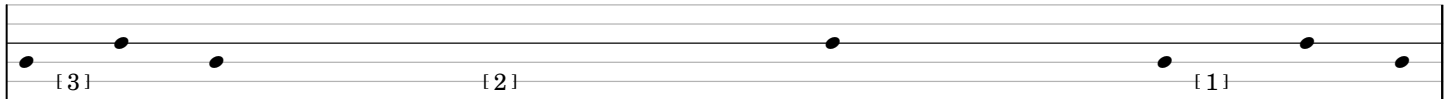
0'24"



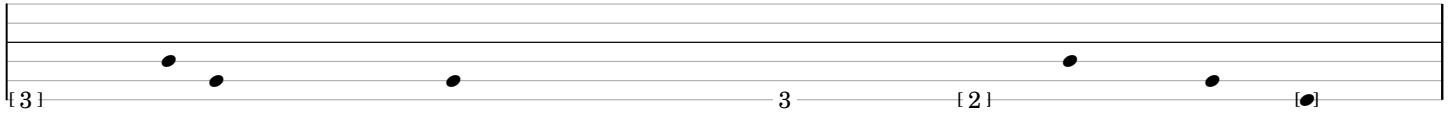
0'36"



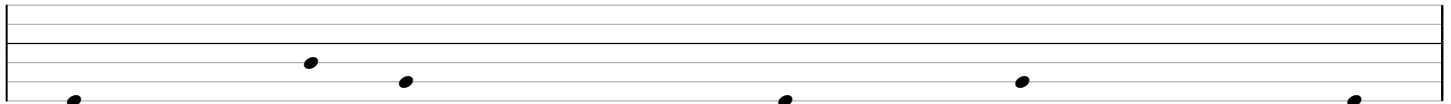
0'48"



1'00"

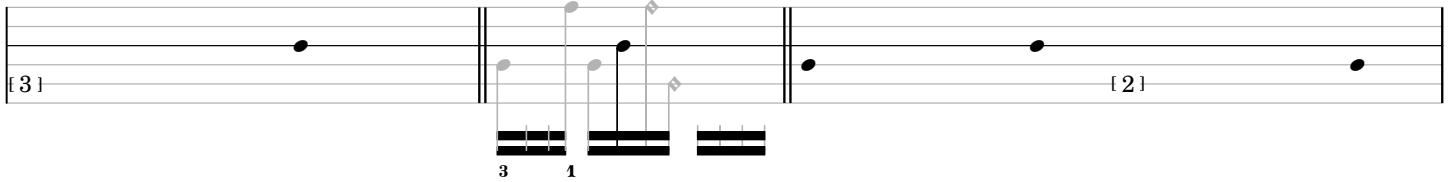


1'12"

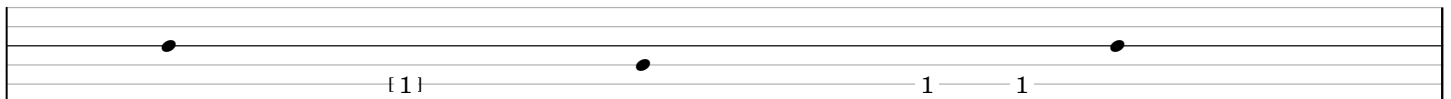


1'24"

1'28"



1'36"



[illegible]

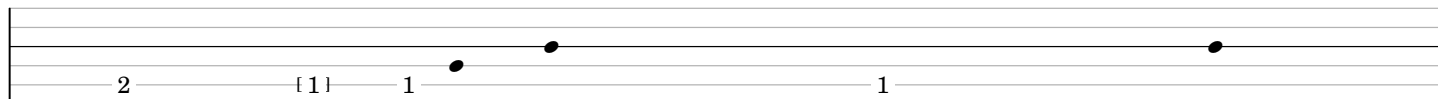
A number line from 0 to 10. The tick mark at 3 is labeled with the number 3. The tick mark at 1 is labeled with the number [3].

The first system of the musical score for 'The Rose Tree' consists of a single staff. It begins with a treble clef and a key signature of one flat (B-flat). The melody starts with a quarter note on G4, followed by a quarter note on F4, and then a quarter rest. This is followed by a measure containing a quarter note on G4, a quarter note on A4, and a quarter note on B4. The system ends with a double bar line.

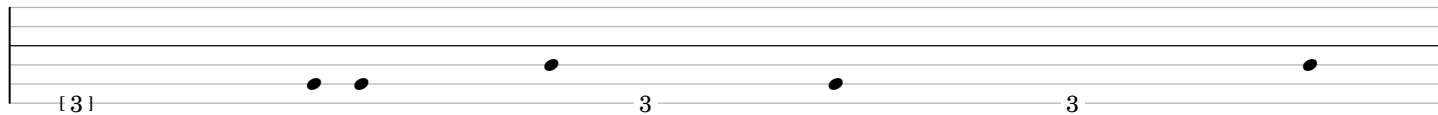
The plot shows the function $f(x) = 2x^2 - 2x + 1$ for $x \in [0, 10]$. The x-axis is labeled from 0 to 10, and the y-axis is labeled from 0 to 10. The function is a parabola opening upwards with its vertex at $(0.5, 0.5)$. The plot includes the function curve and its values at integer points from $x=0$ to $x=10$.

The first system of the musical score consists of a single staff. It begins with a treble clef and a key signature of one flat (B-flat). The notation includes a triplet of eighth notes, followed by a whole note, and then a half note. The system concludes with a double bar line.

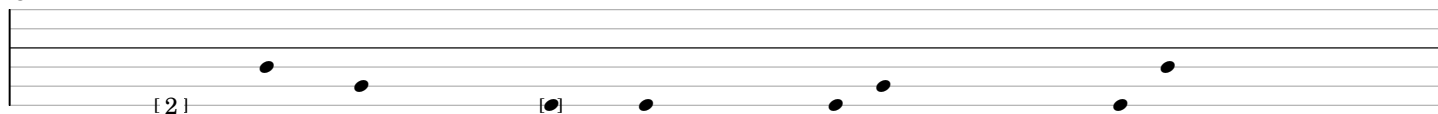
3'24"



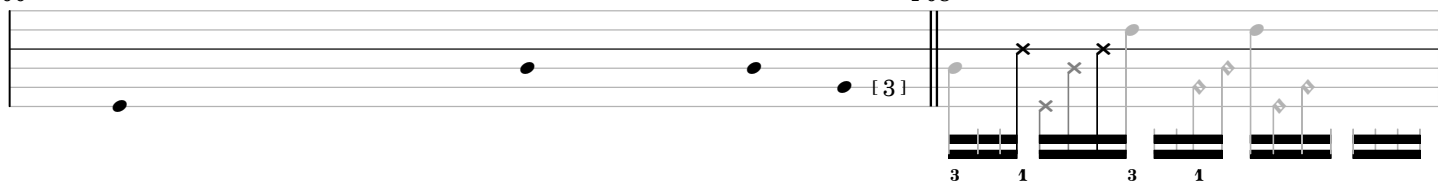
3'36"



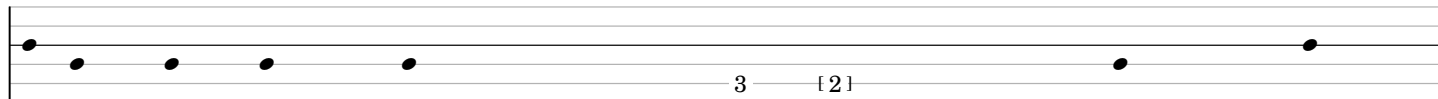
3'48"



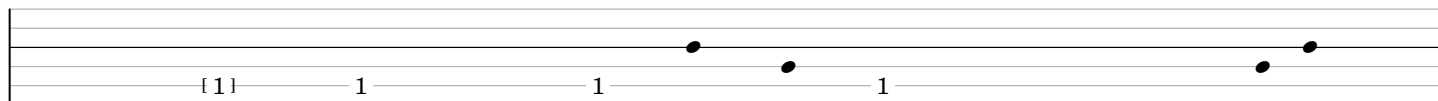
4'00"



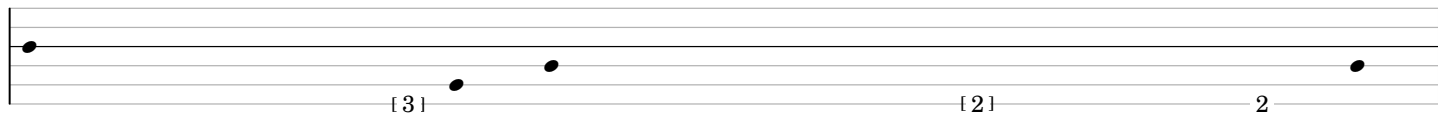
4'12"



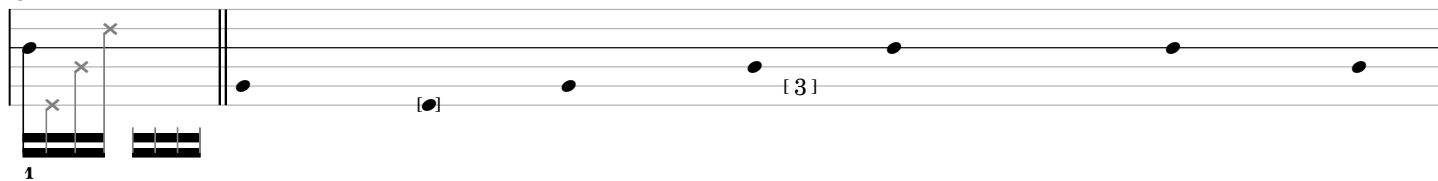
4'24"



4'36"



4'48"



5'00" 5'04"

5'12"

5'24"

5'36" 5'44"

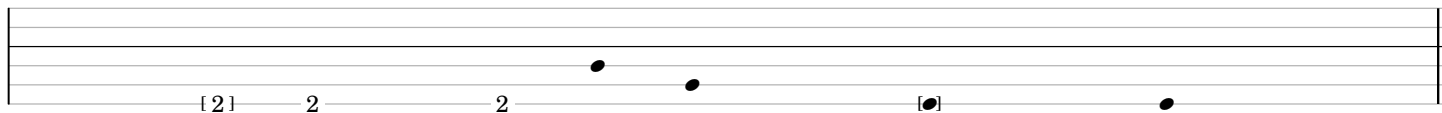
5'48"

6'00"

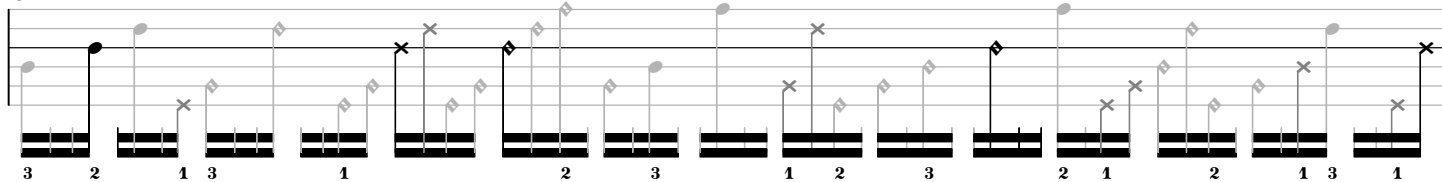
6'12" 6'16"

6'24"

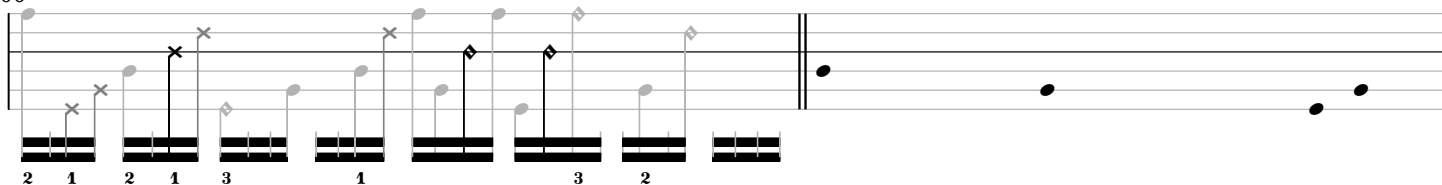
6'36"



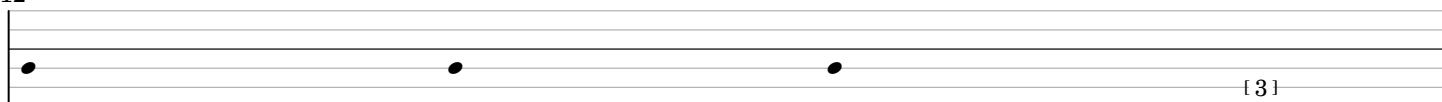
6'48"



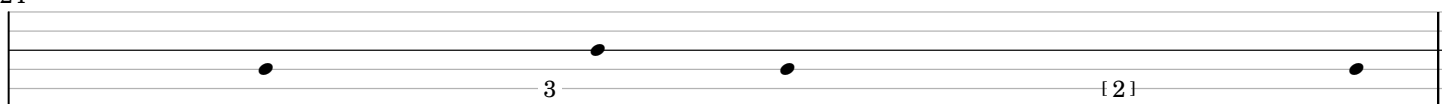
7'00"



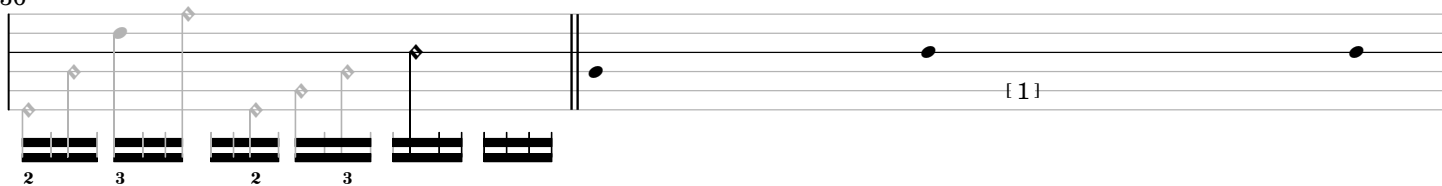
7'12"



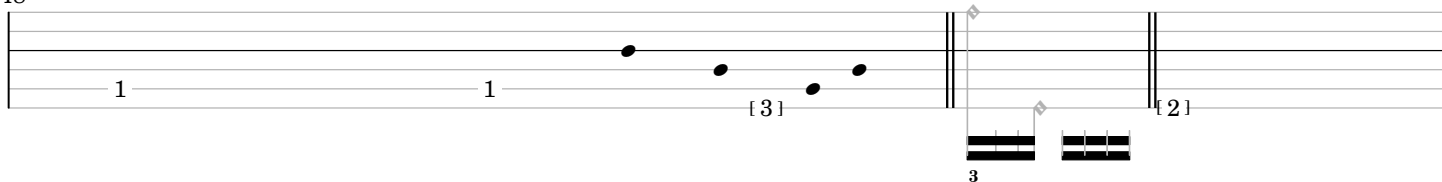
7'24"



7'36"

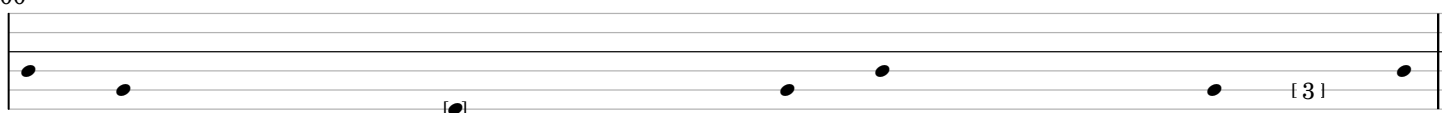


7'48"



7'56"

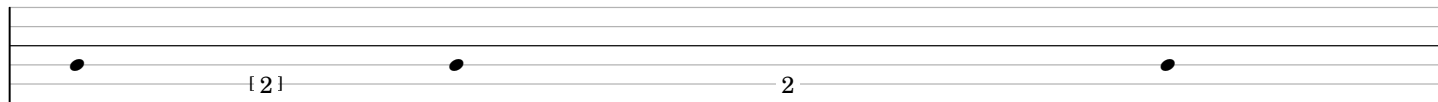
8'00"



8'12"

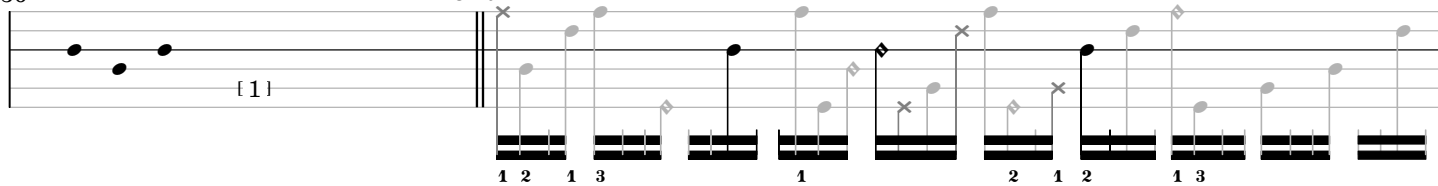


8'24"

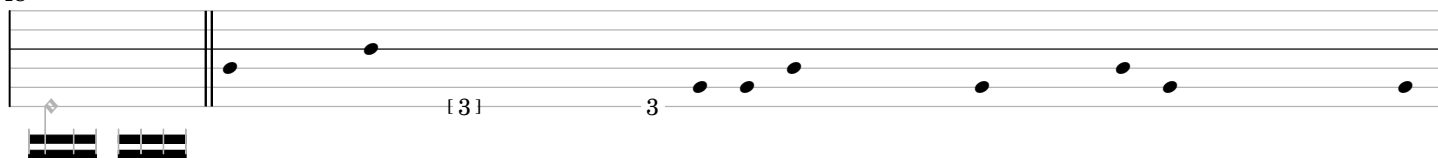


8'36"

8'40"

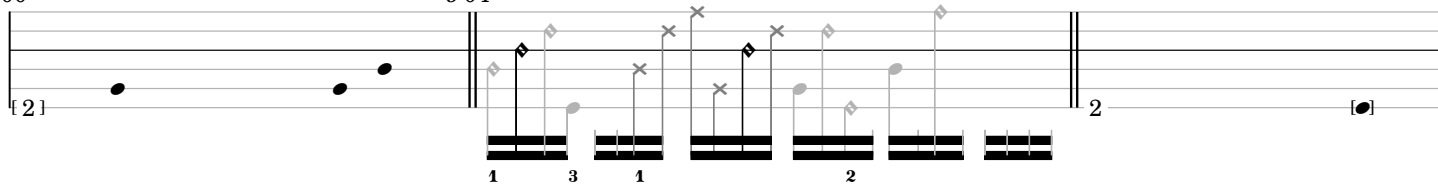


8'48"

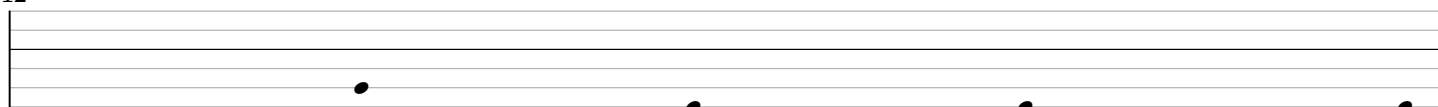


9'00"

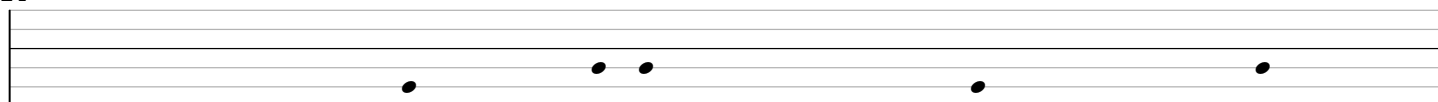
9'04"



9'12"



9'24"

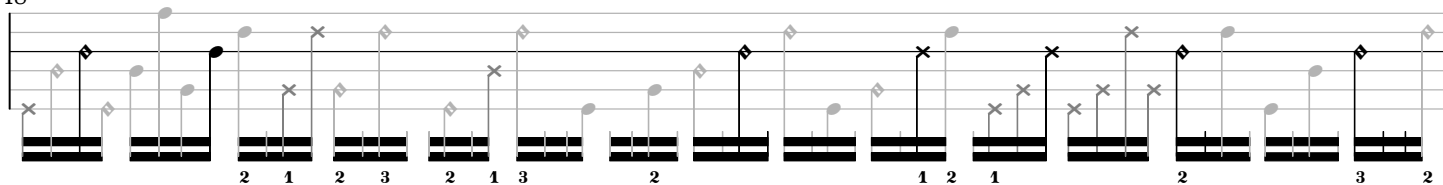


9'36"

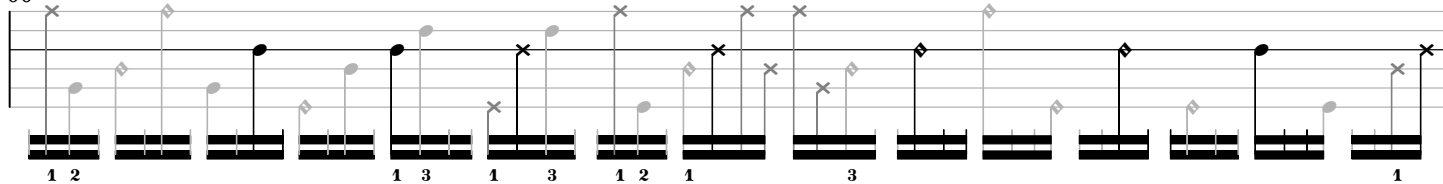
9'44"



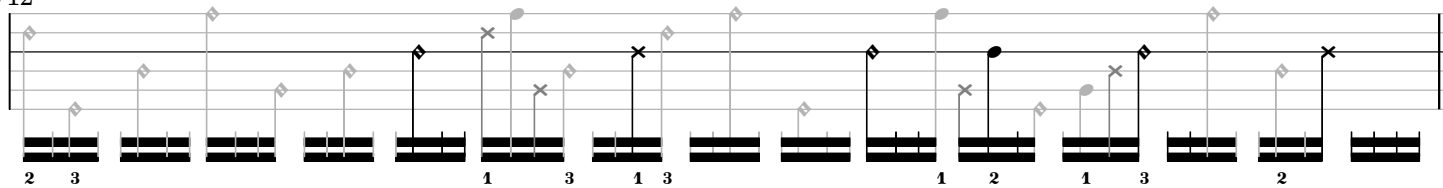
9'48"



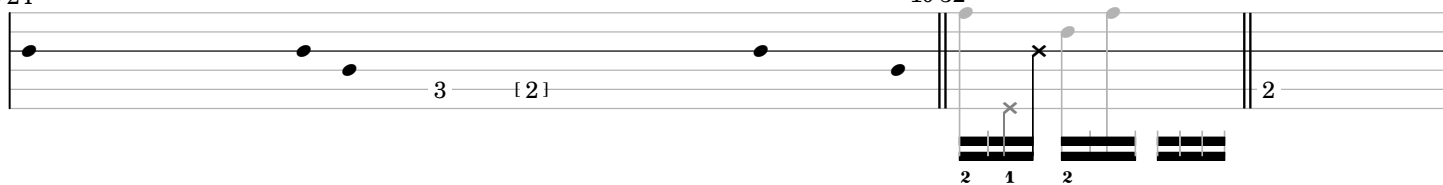
10'00"



10'12"

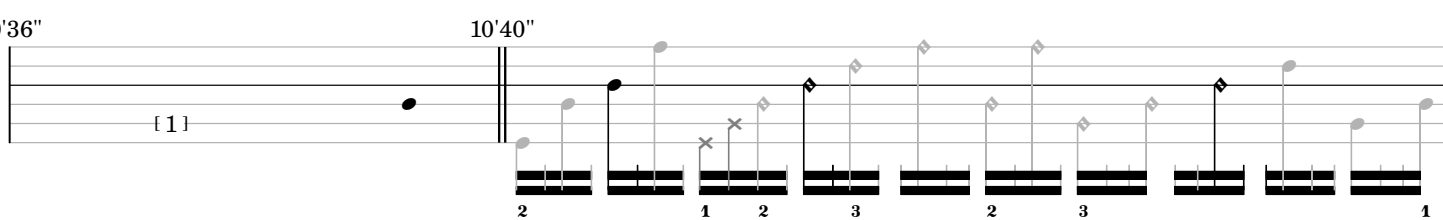


10'24"



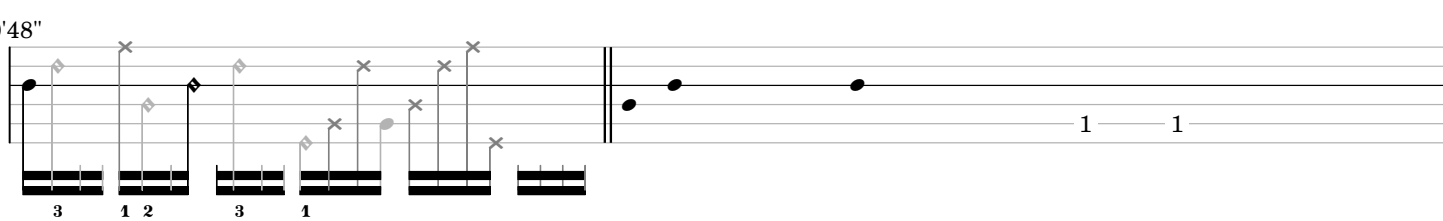
10'32"

10'36"

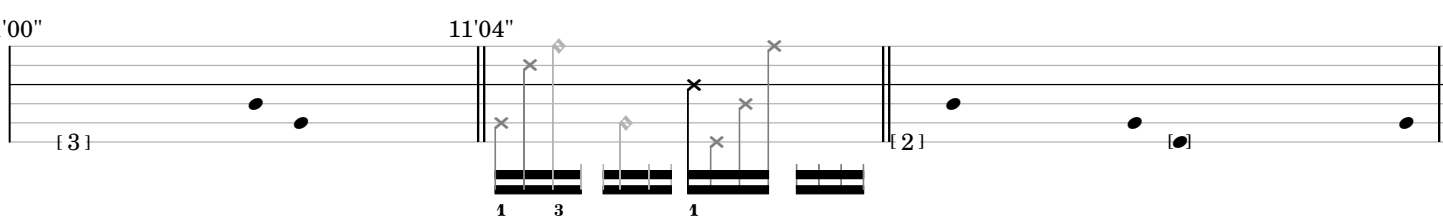


10'40"

10'48"

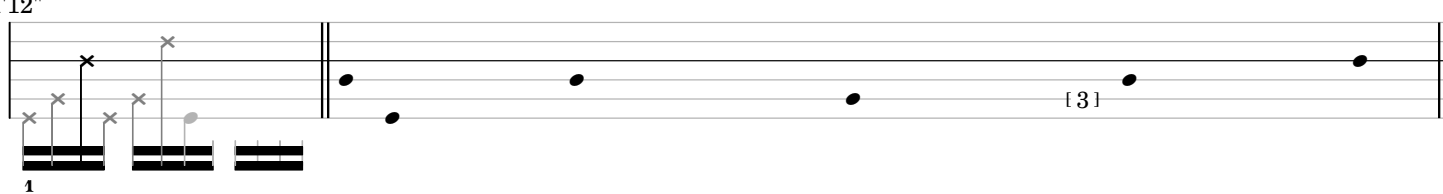


11'00"



11'04"

11'12"



11'24"

2 1 3 2 3 2 1 3 1 2 1

11'36" 11'44"

1 2 1 1 3 1

11'48"

3 [2]

12'00"

1 2 1 2 1 2 1 2 2 2 1 2 1 2

12'12"

[3]

12'24"

2 1

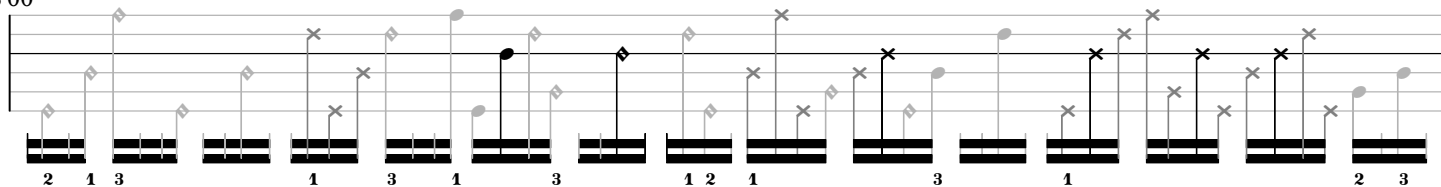
12'36"

3 1 2 [1] 1

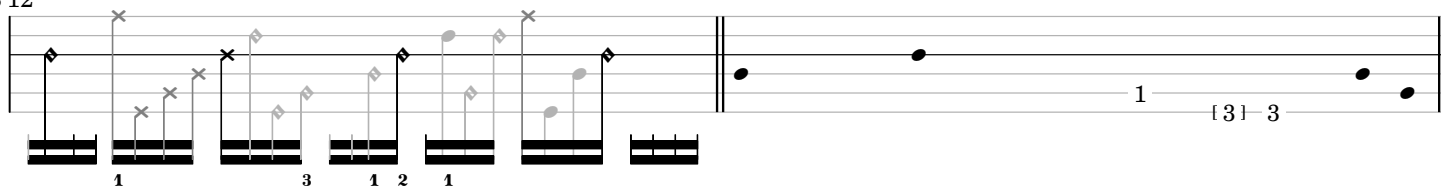
12'48"

2 3 1 2 1 3 1 3 1 3 1 3

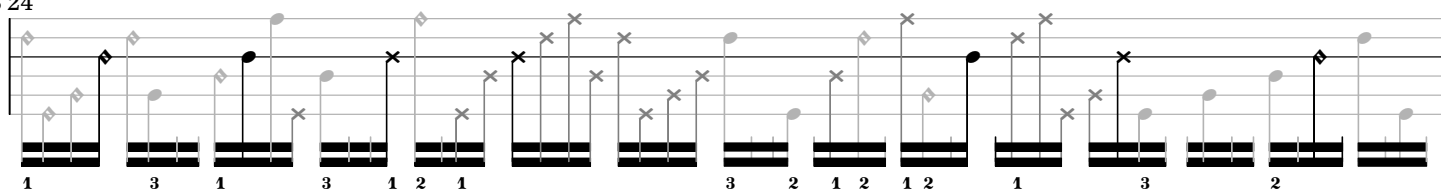
13'00"



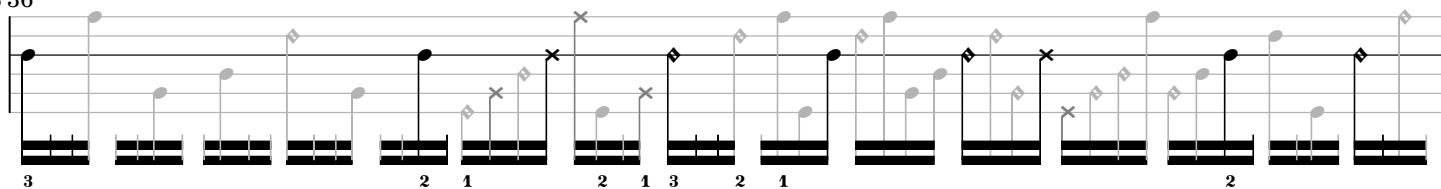
13'12"



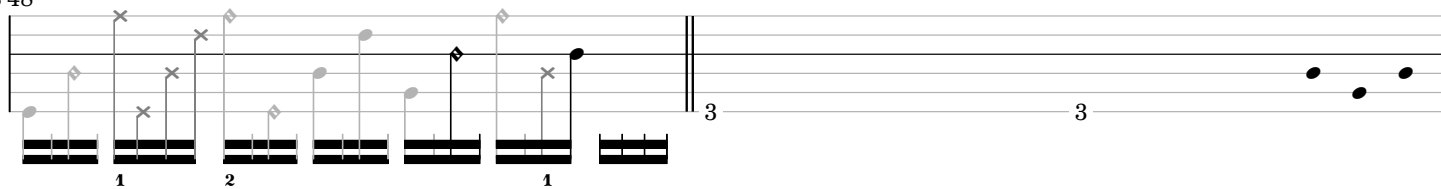
13'24"



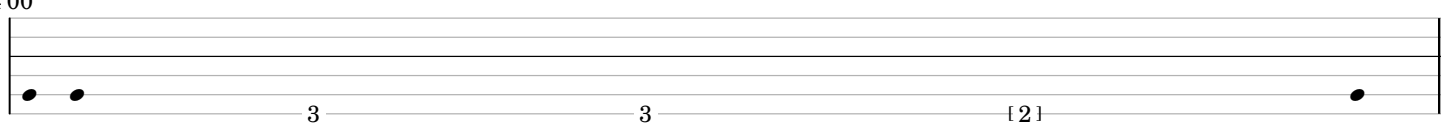
13'36"



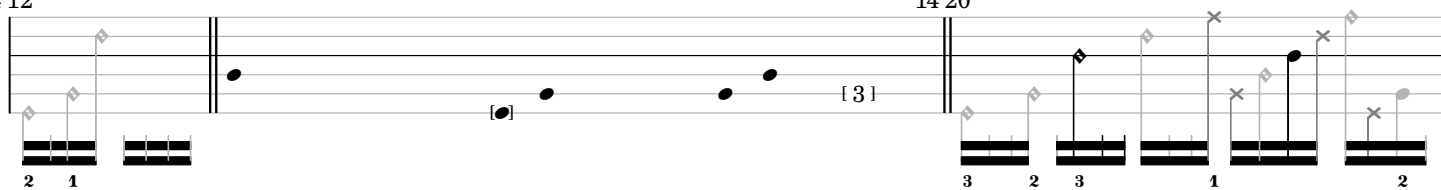
13'48"



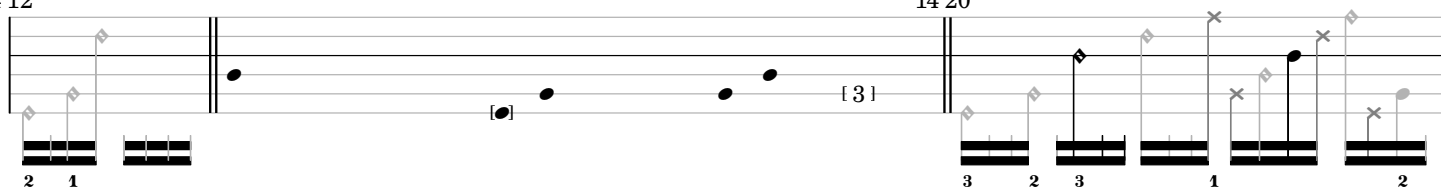
14'00"



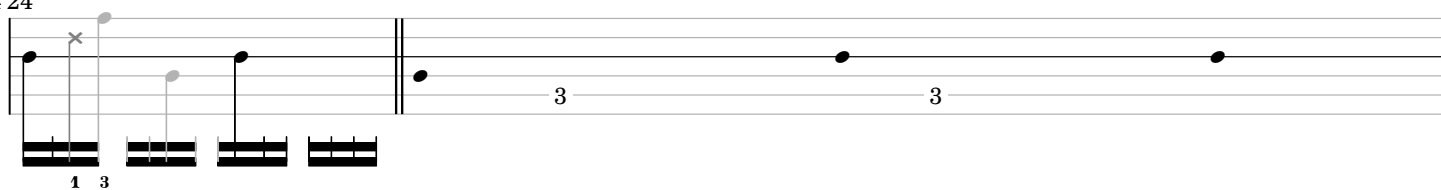
14'12"



14'20"



14'24"



14'36"

14'44"

3 [2]

3 1 2 3 1 2

14'48"

1 2 1 2 3 2 3 1 2 1 2 3 2 1

15'00"

3 2 1 2 1 2 3 1 3 2 1 3 3 2

15'12"

3 1 2 1 3 2 3 2

[1] 1

15'24"

2 1 2 1 2 1 3 1 2 1

[3]

15'36"

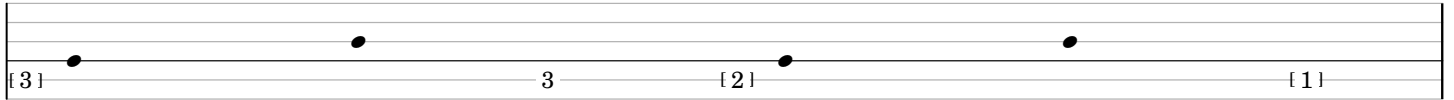
[2] 2 [6]

ostinato and interrupt

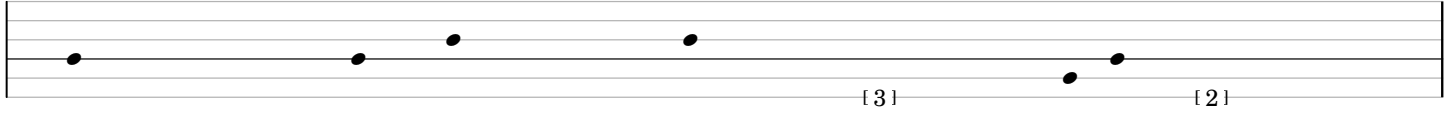
ensemble part 4

michael winter (mexico city, mx; 2017)
version generated: 2017.12.06

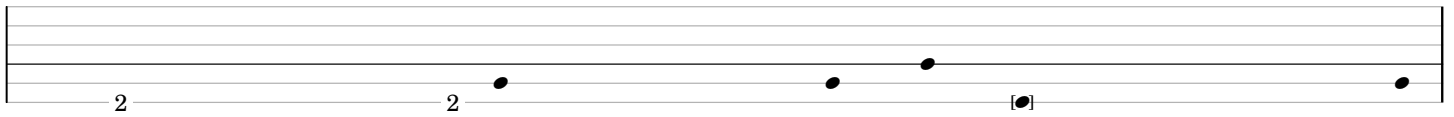
0'12"



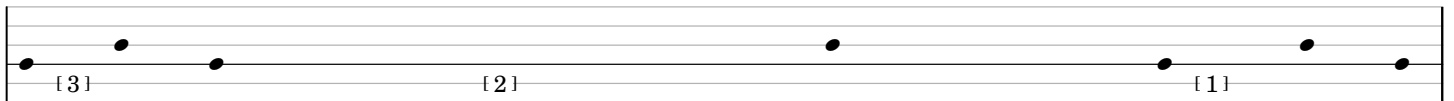
0'24"



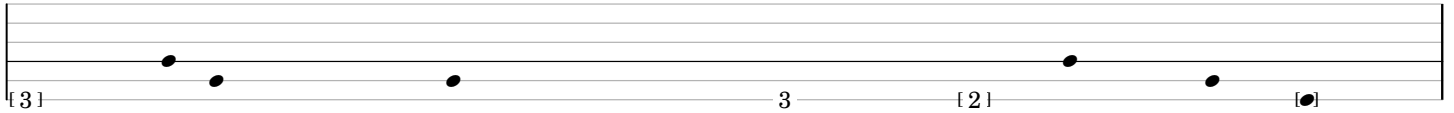
0'36"



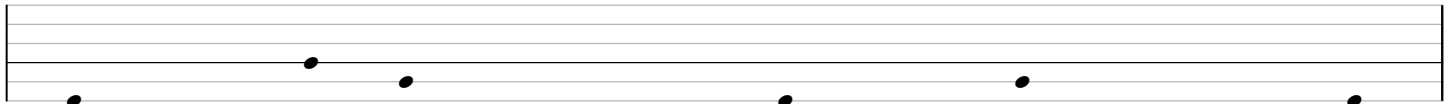
0'48"



1'00"

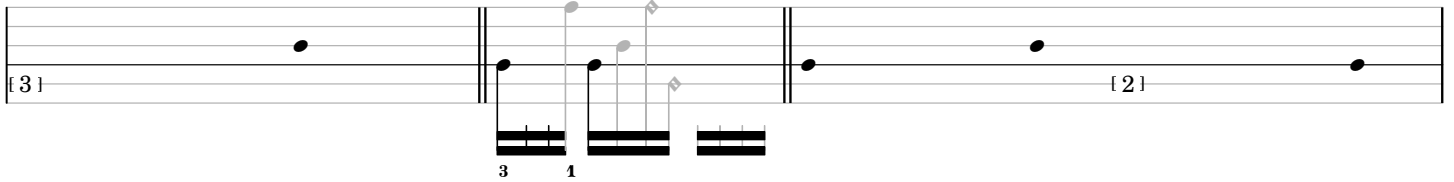


1'12"

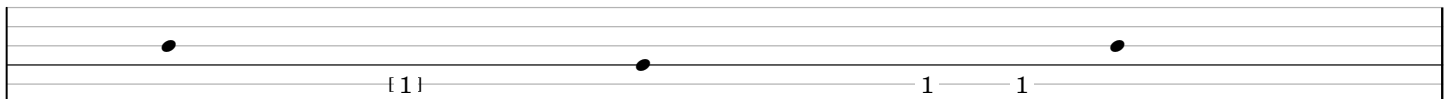


1'24"

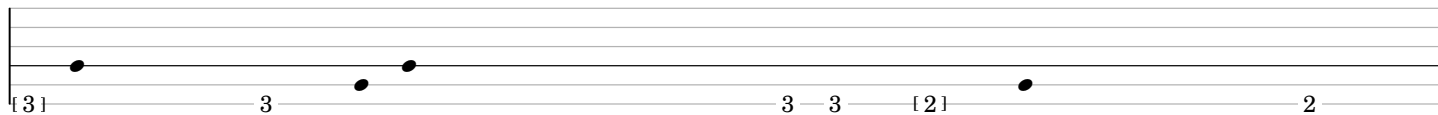
1'28"



1'36"

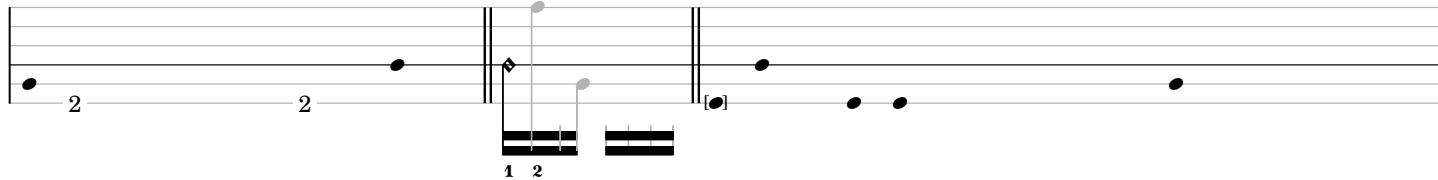


1'48"

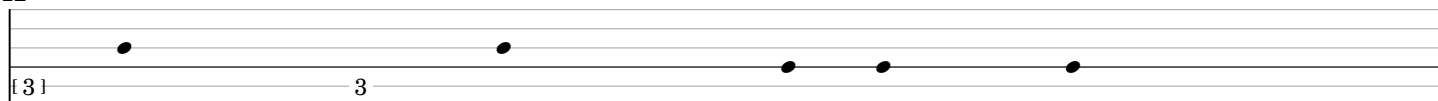


2'00"

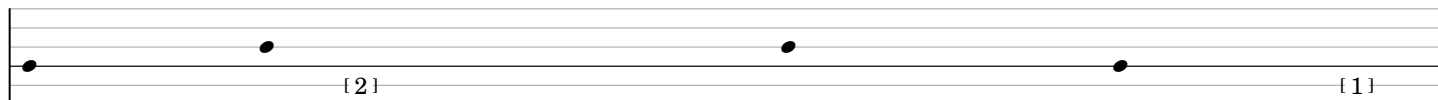
2'04"



2'12"

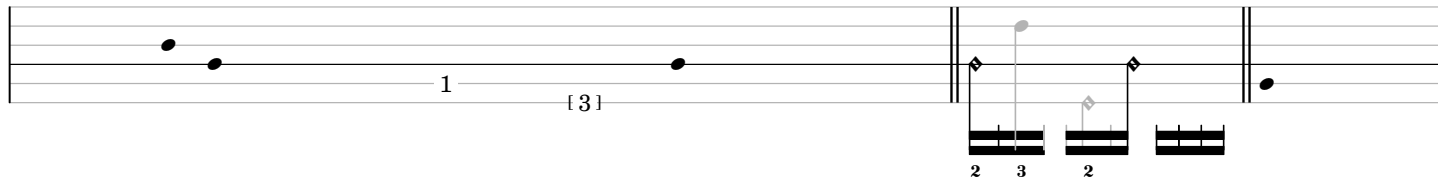


2'24"

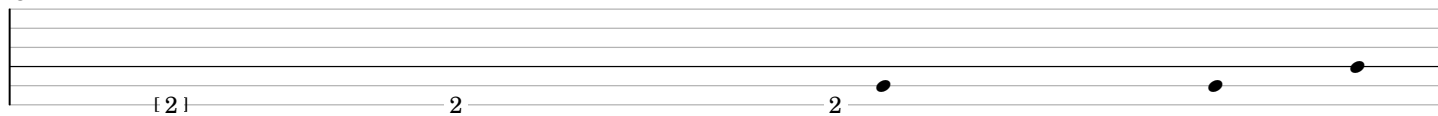


2'36"

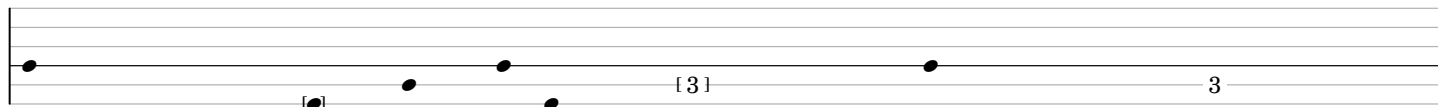
2'44"



2'48"

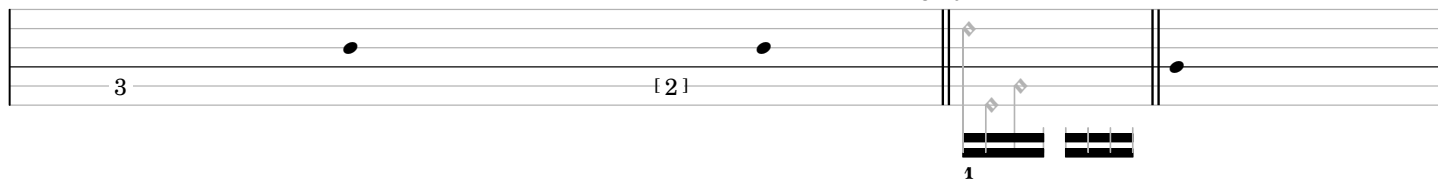


3'00"

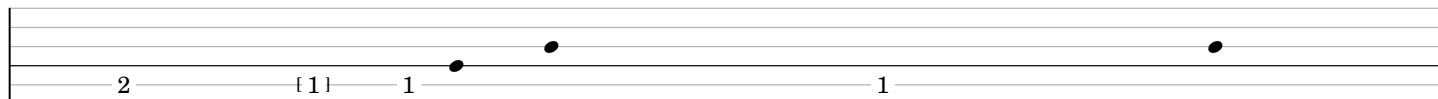


3'12"

3'20"



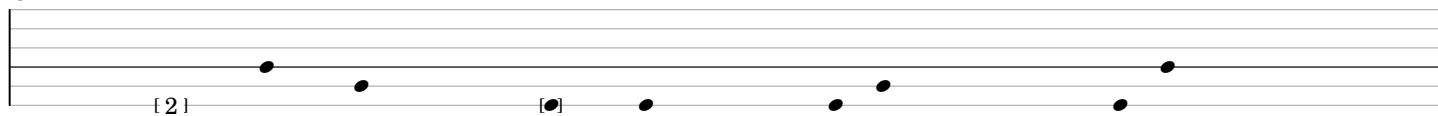
3'24"



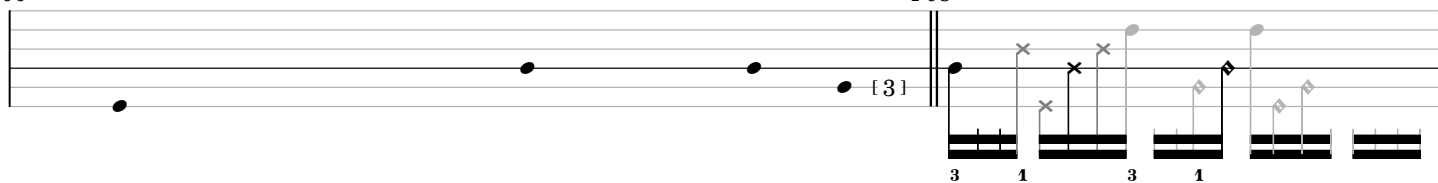
3'36"



3'48"

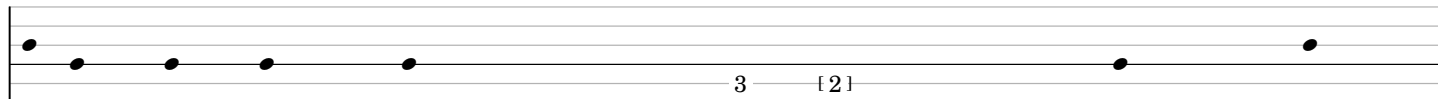


4'00"



4'08"

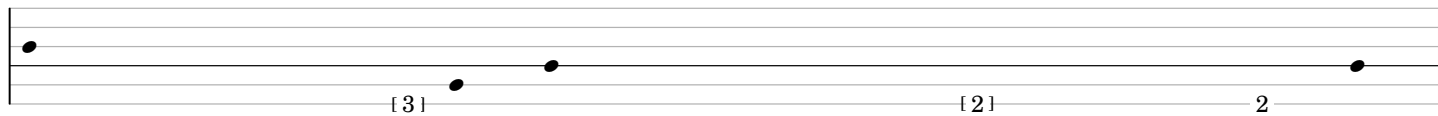
4'12"



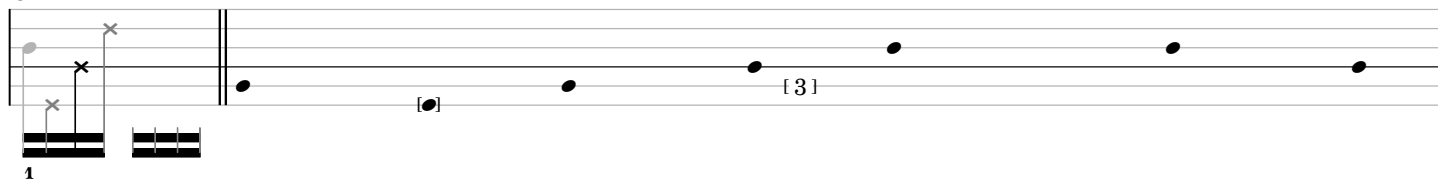
4'24"



4'36"



4'48"



5'00" 5'04"

5'12"

5'24"

5'36" 5'44"

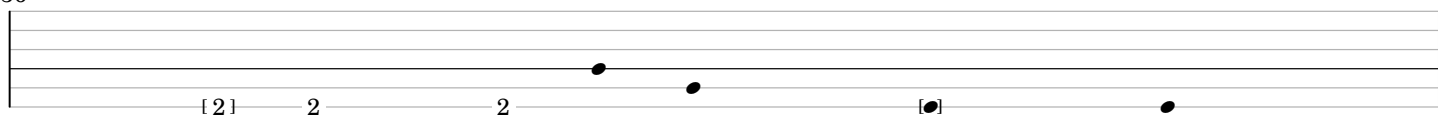
5'48"

6'00"

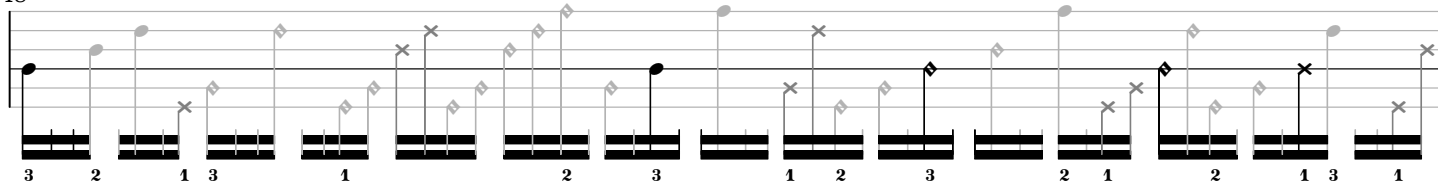
6'12" 6'16"

6'24"

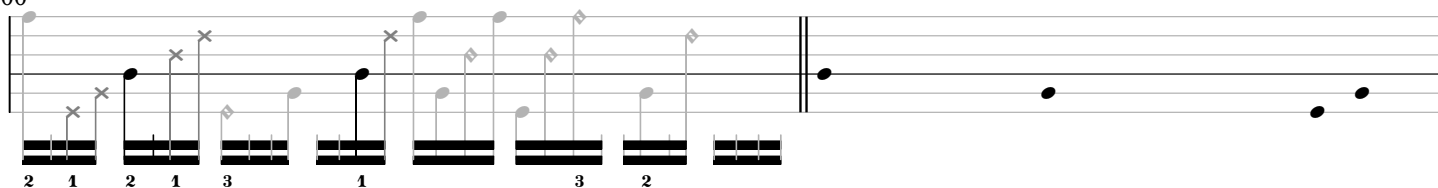
6'36"



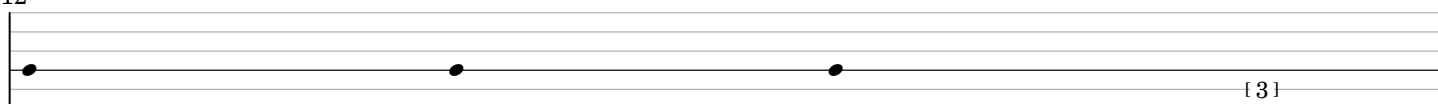
6'48"



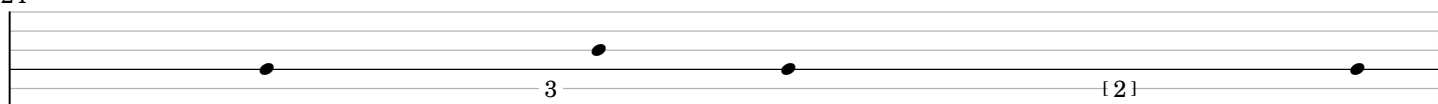
7'00"



7'12"



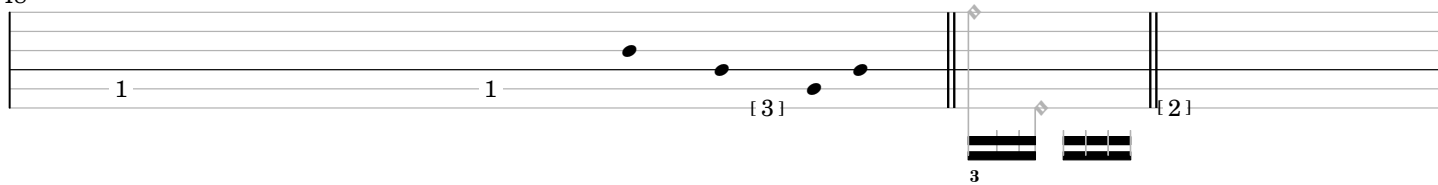
7'24"



7'36"

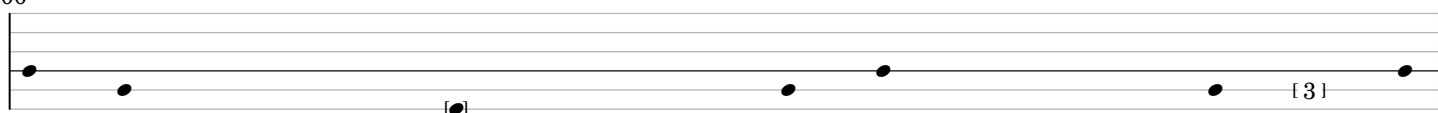


7'48"

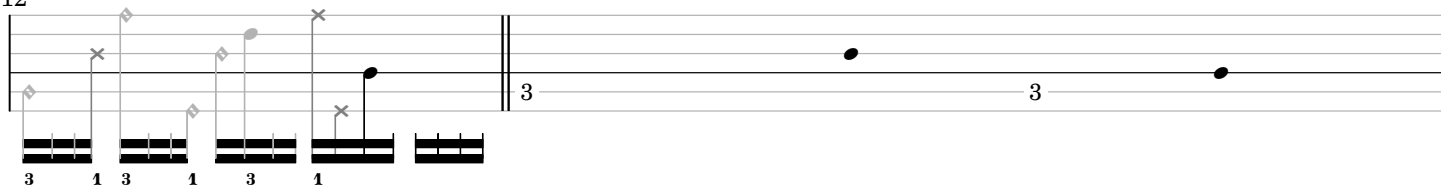


7'56"

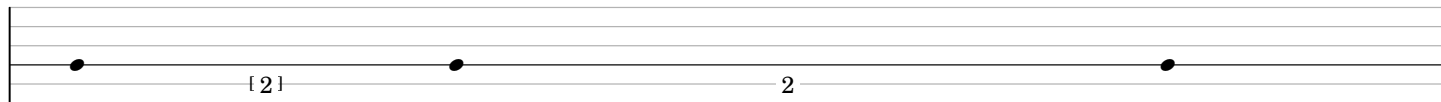
8'00"



8'12"

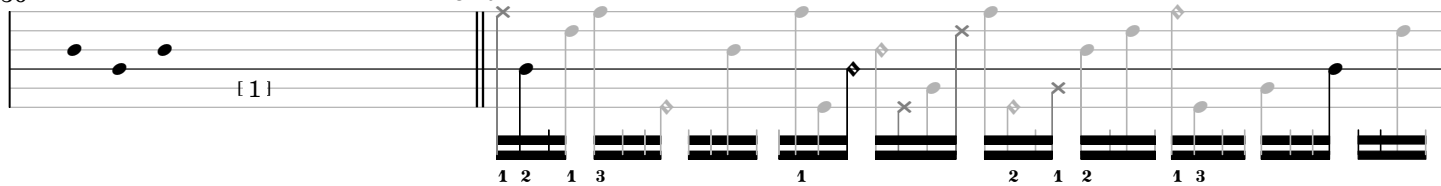


8'24"

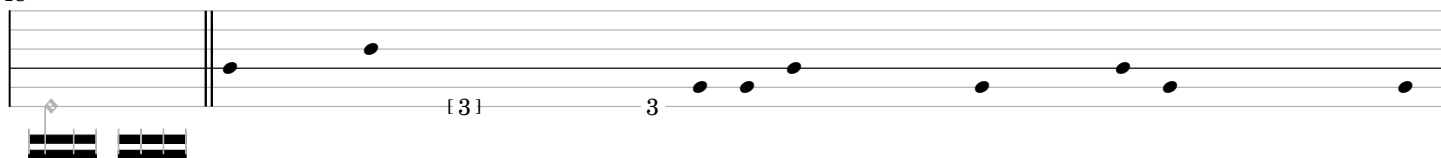


8'36"

8'40"

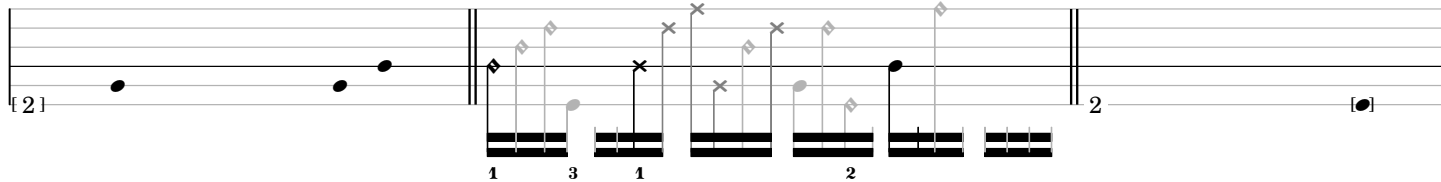


8'48"

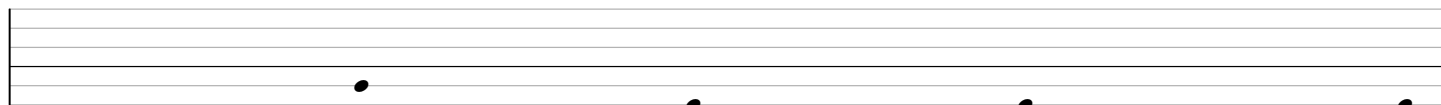


9'00"

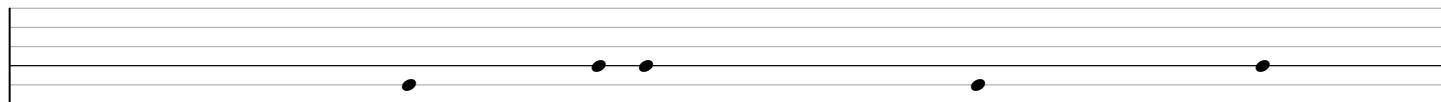
9'04"



9'12"



9'24"

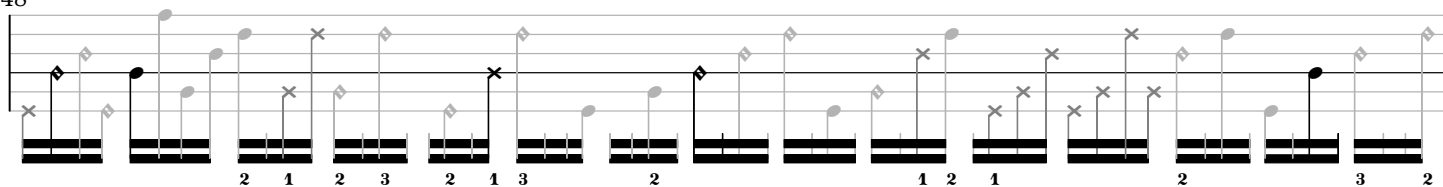


9'36"

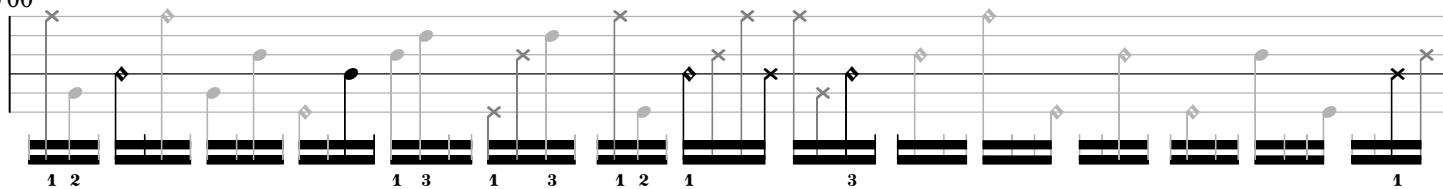
9'44"



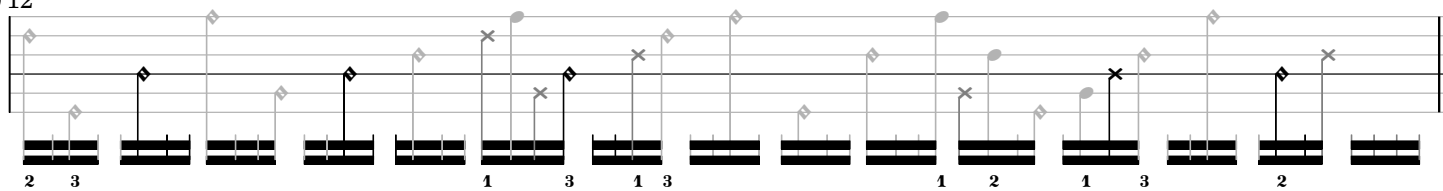
9'48"



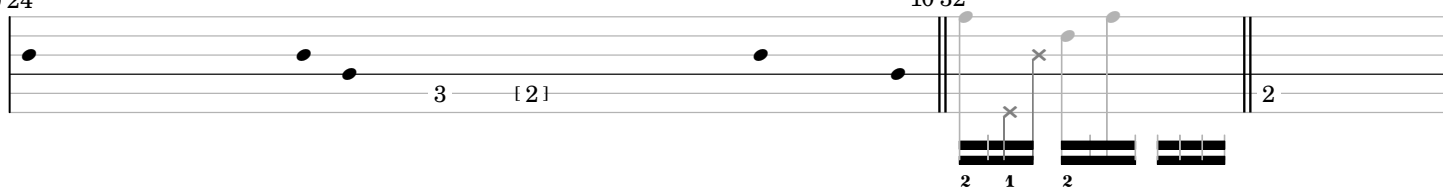
10'00"



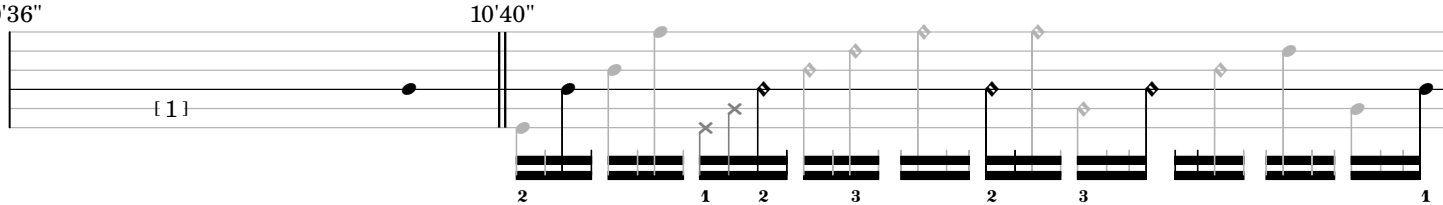
10'12"



10'24"

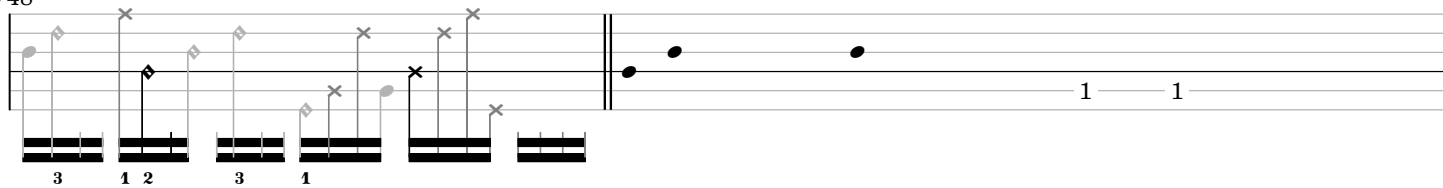


10'36"

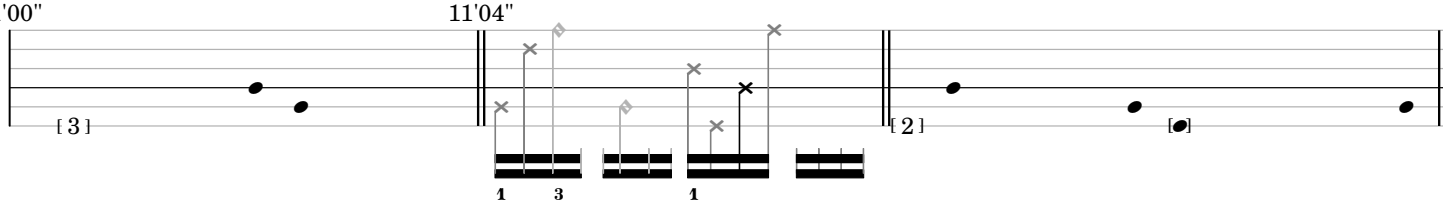


10'40"

10'48"

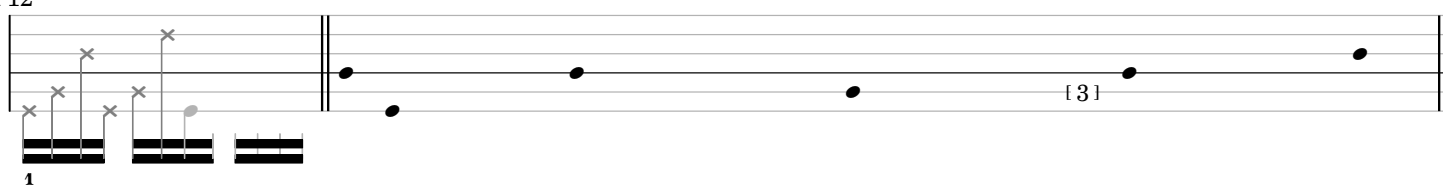


11'00"



11'04"

11'12"



11'24"

The musical score for 'The Rose Tree' is presented on a single staff. The melody is written in a treble clef with a key signature of one flat (B-flat). The tempo is marked 'Allegretto' and the time signature is 3/4. The score consists of two systems. The first system contains 11 measures, and the second system contains 3 measures. The melody is primarily composed of eighth and sixteenth notes, with some rests. There are several accidentals, including flats and naturals. The score is accompanied by a piano accompaniment consisting of a continuous eighth-note pattern in the right hand and a simple bass line in the left hand. The piano part is marked with a 'p' (piano) dynamic. The score ends with a double bar line and a repeat sign.

11'36"

The first system of the musical score for 'The Rose Tree' consists of two staves. The upper staff is a treble clef with a key signature of one flat (B-flat). It contains a melody with notes G4, A4, Bb4, A4, G4, F4, E4, D4, C4, and Bb3. The lower staff is a bass clef with a key signature of one flat (B-flat). It contains a bass line with notes G3, A3, Bb3, A3, G3, F3, E3, D3, C3, and Bb2. The system is divided into two measures by a double bar line. The first measure contains the notes G4, A4, Bb4, A4, G4, F4, E4, D4, C4, and Bb3. The second measure contains the notes G3, A3, Bb3, A3, G3, F3, E3, D3, C3, and Bb2. The system is numbered 10 and 11 at the top.

11'44"

11'48"

12'00"

12'12"

[3]

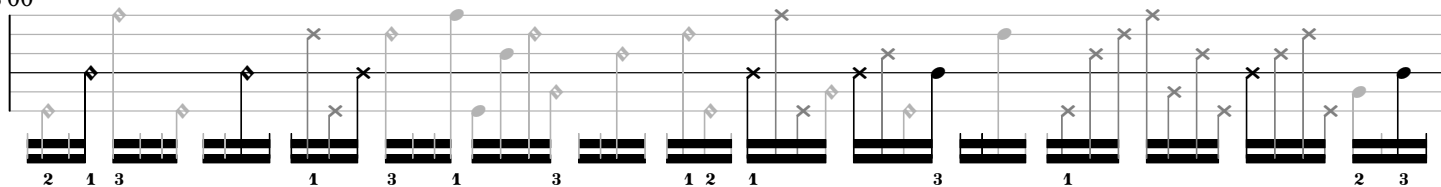
12'24"

[illegible]

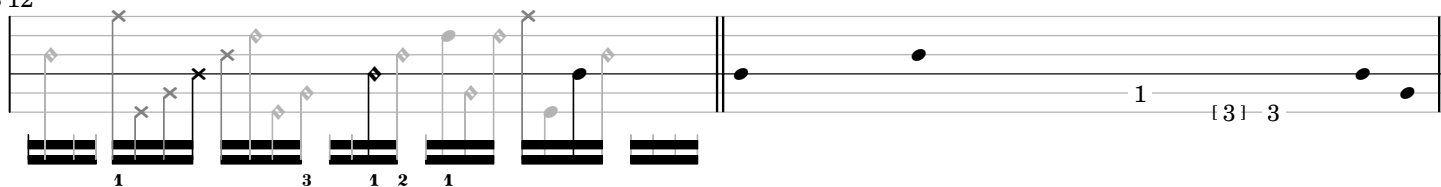
12'36"

12'48"

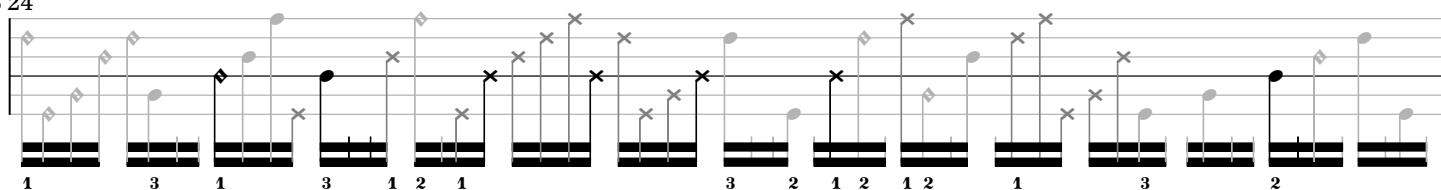
13'00"



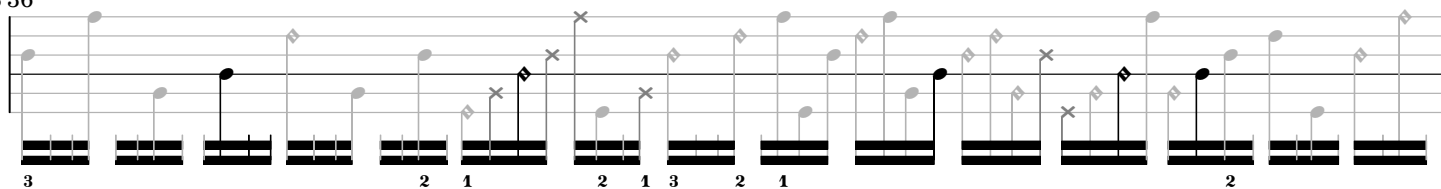
13'12"



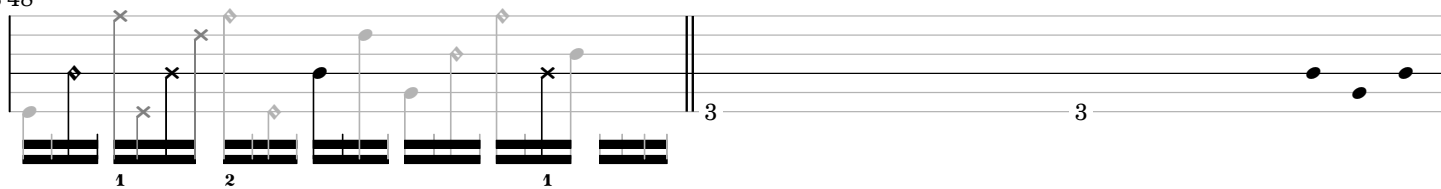
13'24"



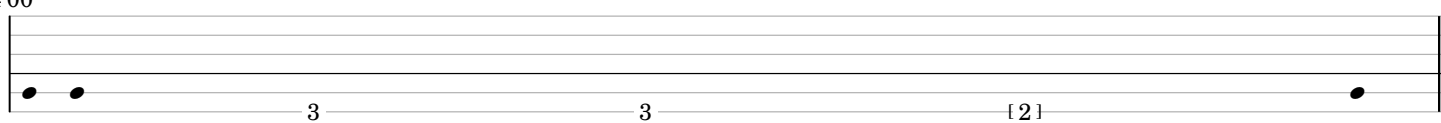
13'36"



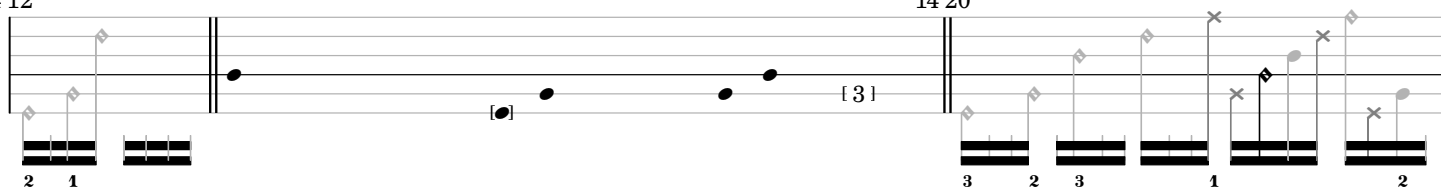
13'48"



14'00"

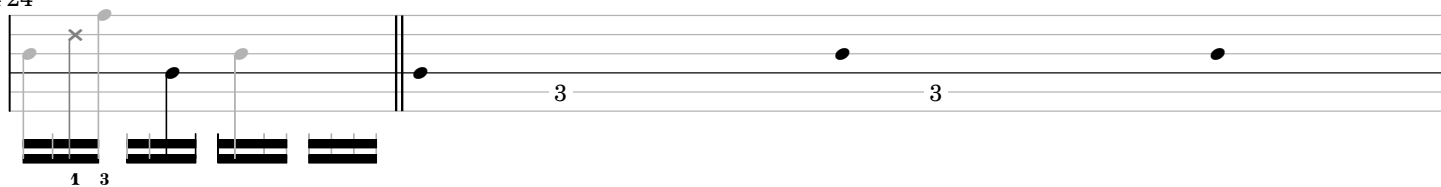


14'12"



14'20"

14'24"



14'36" 14'44"

3 [2]

3 1 2 3 1 2

14'48"

1 2 1 2 3 2 3 1 2 1 2 3 2 1

15'00"

3 2 1 2 1 2 3 1 3 2 1 3 2

15'12"

3 1 2 1 3 2 3 2 [1] 1

15'24"

2 1 2 1 2 1 3 1 2 1 3 [3]

15'36"

[2] 2 [2]

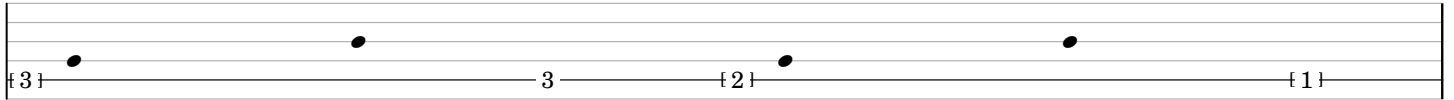
ostinato and interrupt

ensemble part 5

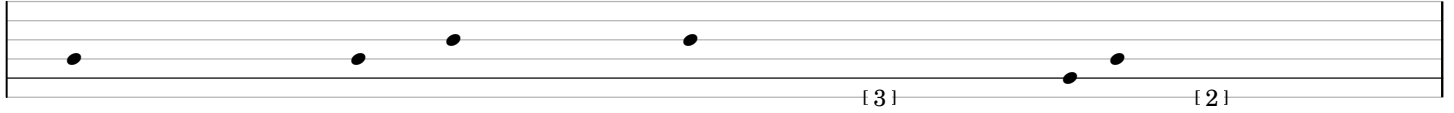
michael winter (mexico city, mx; 2017)

version generated: 2017.12.06

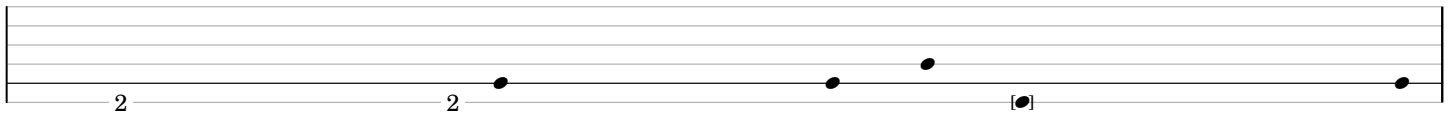
0'12"



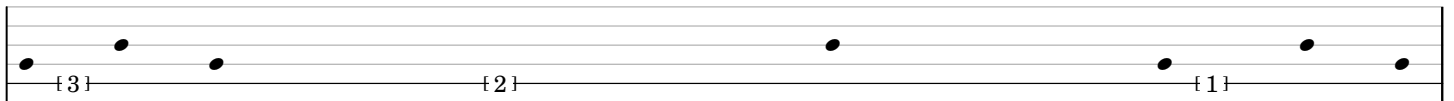
0'24"



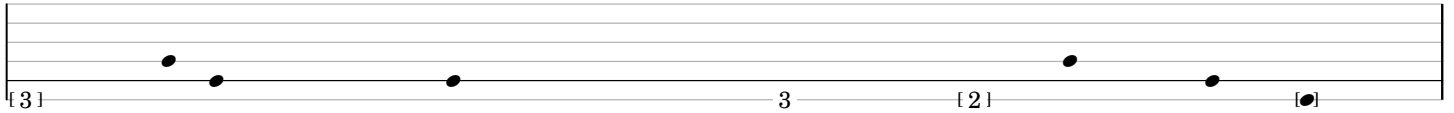
0'36"



0'48"



1'00"

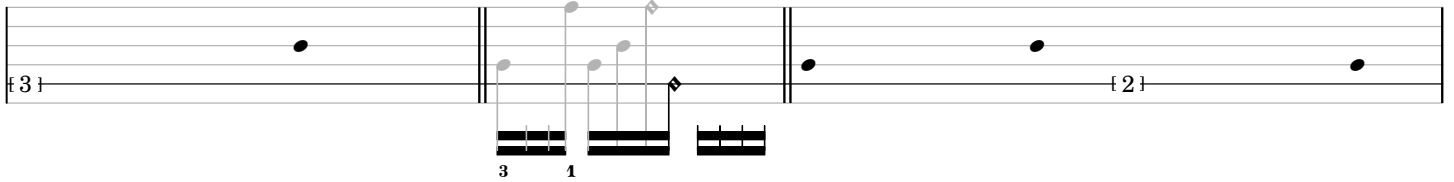


1'12"

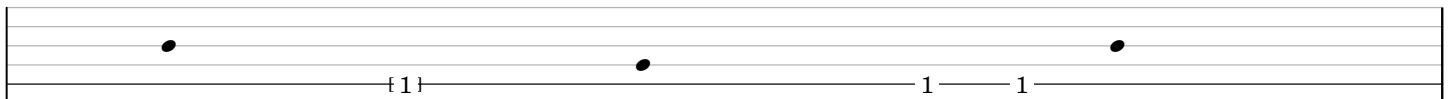


1'24"

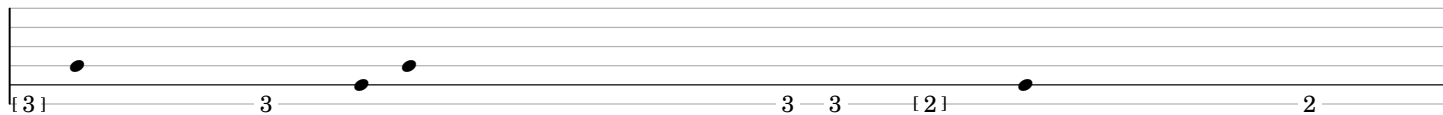
1'28"



1'36"

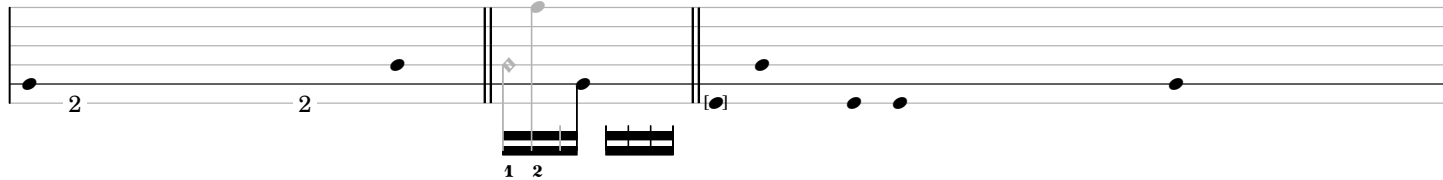


1'48"

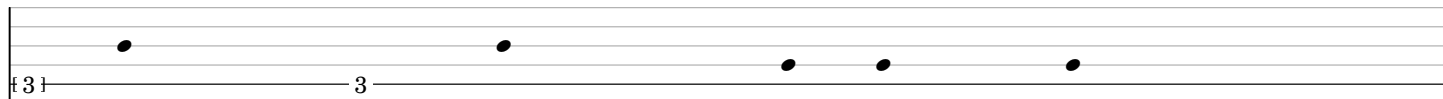


2'00"

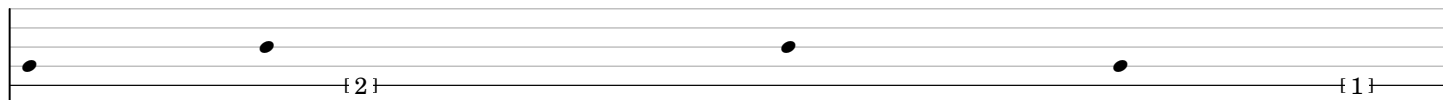
2'04"



2'12"

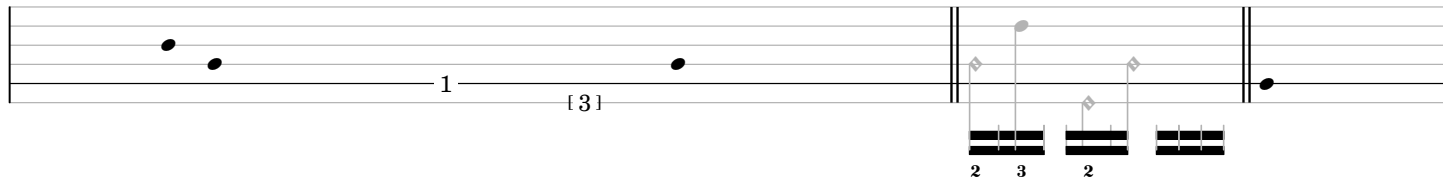


2'24"

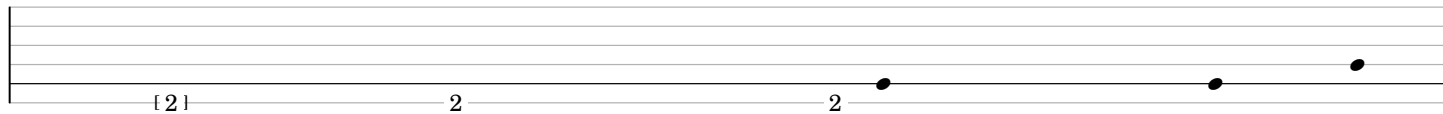


2'36"

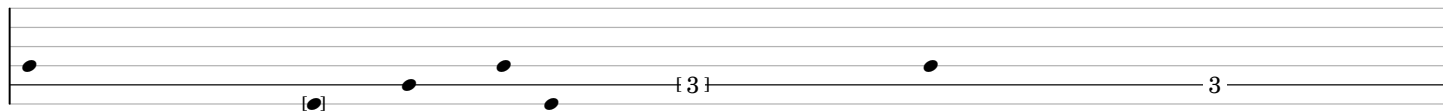
2'44"



2'48"

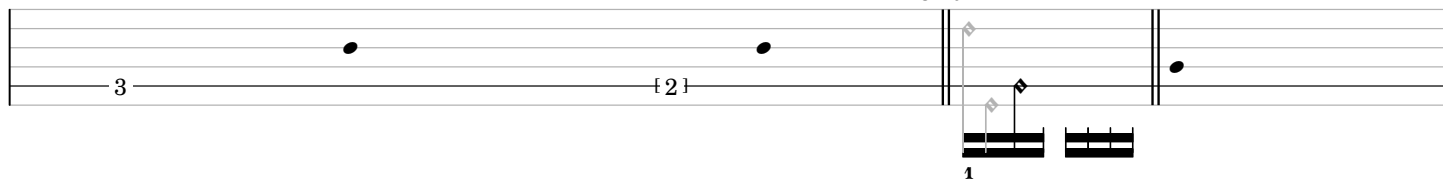


3'00"

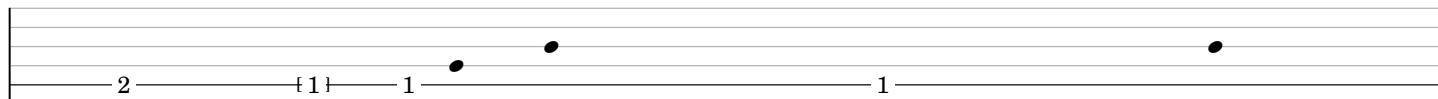


3'12"

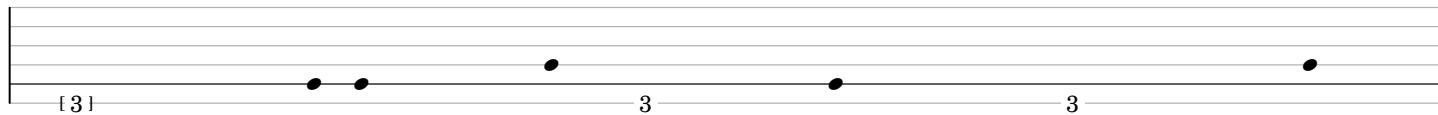
3'20"



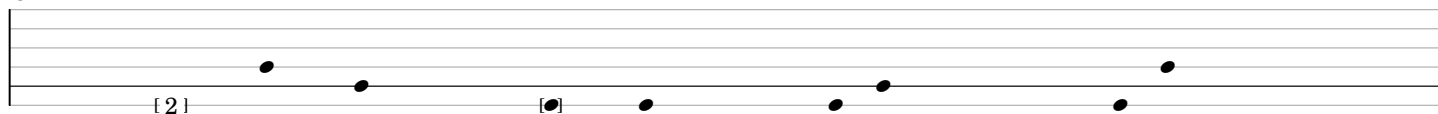
3'24"



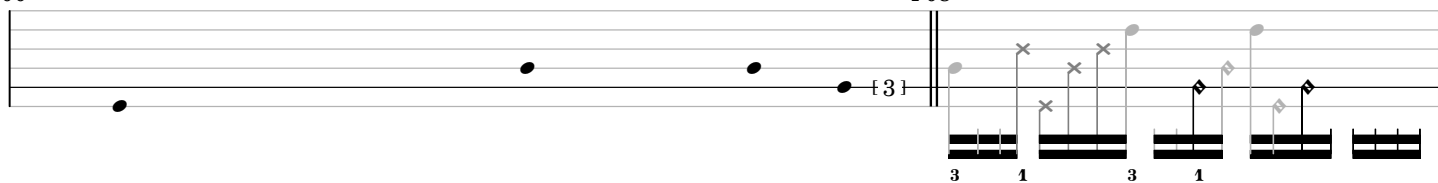
3'36"



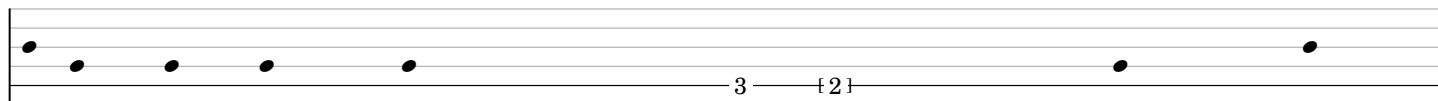
3'48"



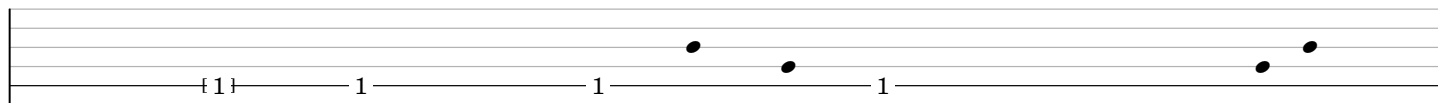
4'00"



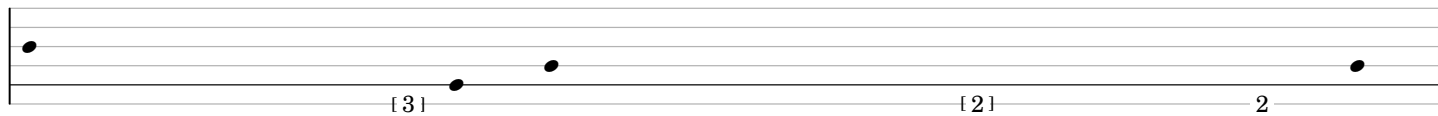
4'12"



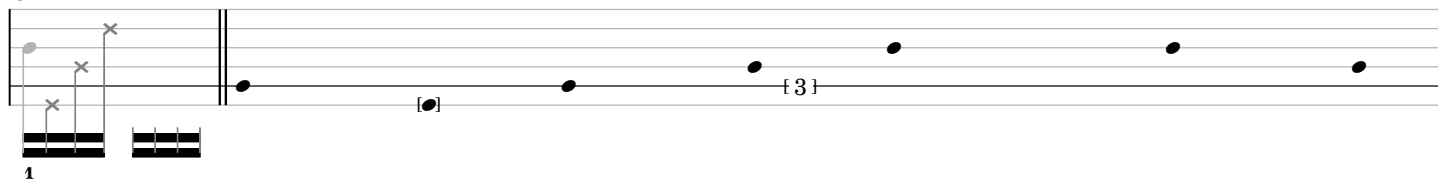
4'24"



4'36"



4'48"



5'00" 5'04"

5'12"

5'24"

5'36" 5'44"

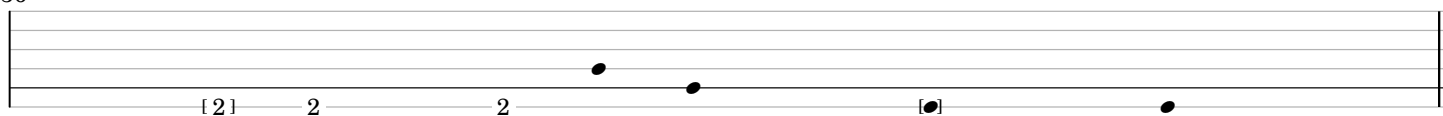
5'48"

6'00"

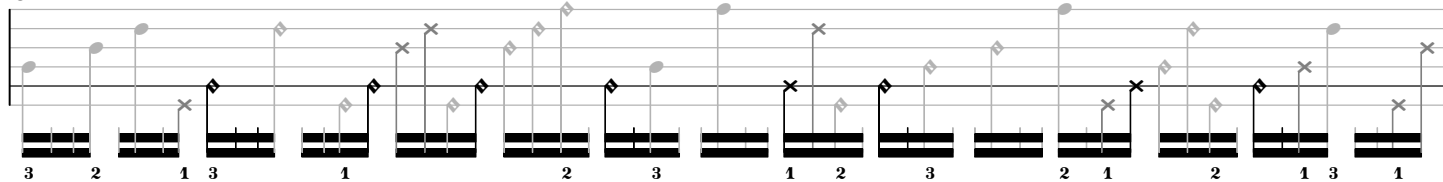
6'12" 6'16"

6'24"

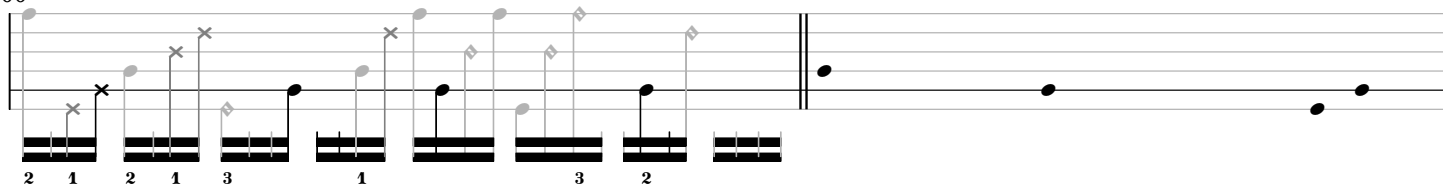
6'36"



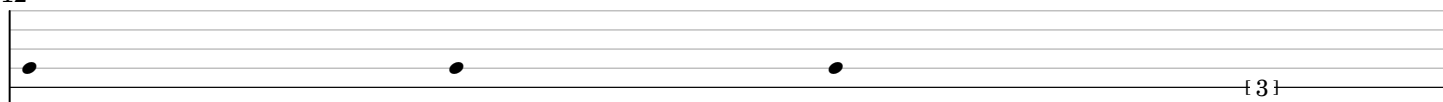
6'48"



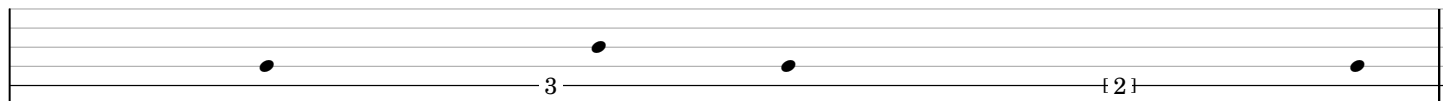
7'00"



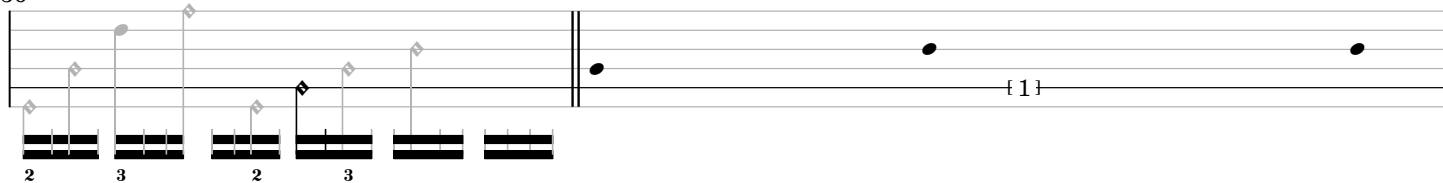
7'12"



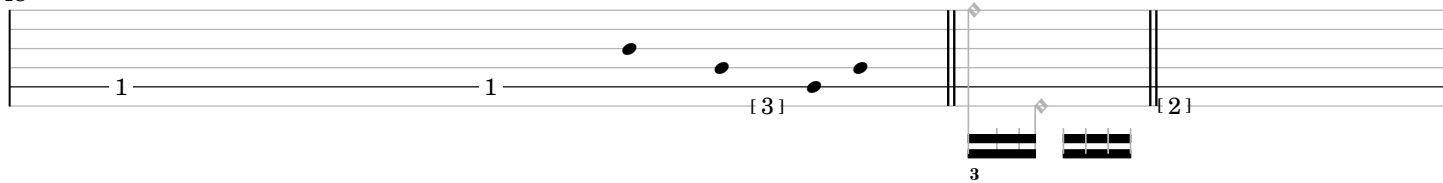
7'24"



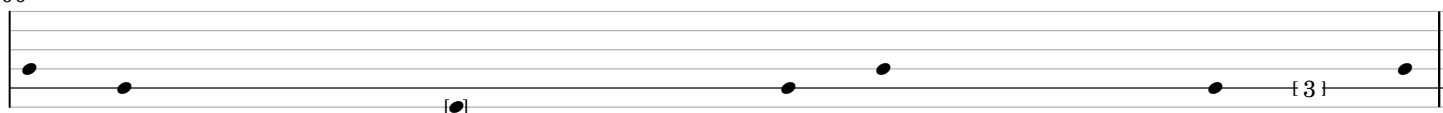
7'36"



7'48"



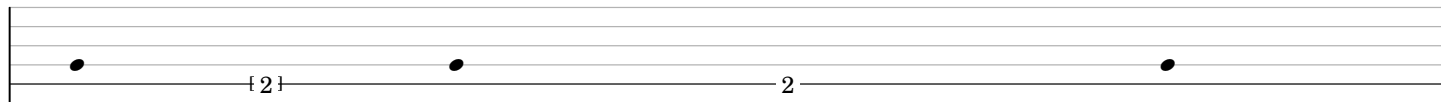
8'00"



8'12"

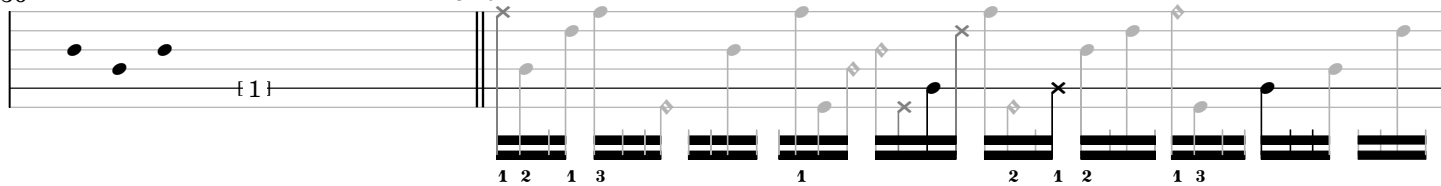


8'24"

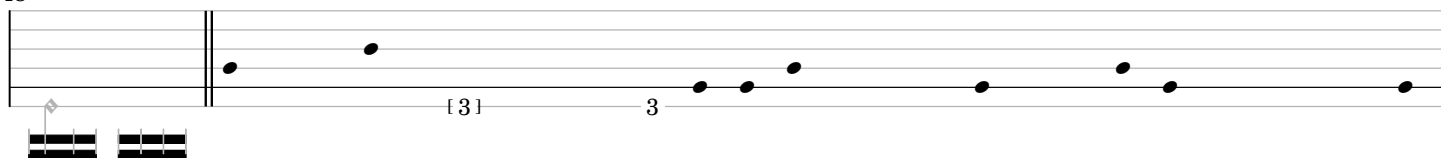


8'36"

8'40"

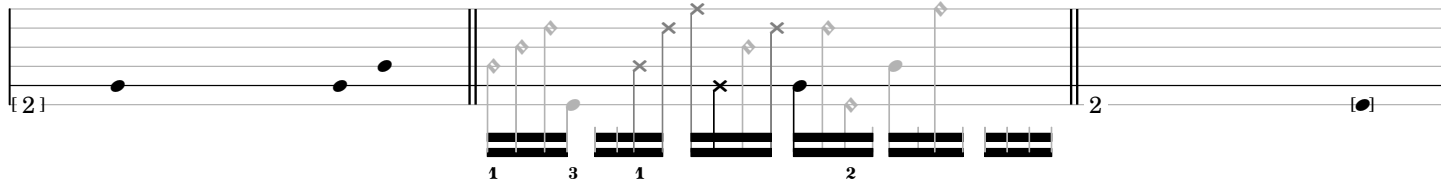


8'48"

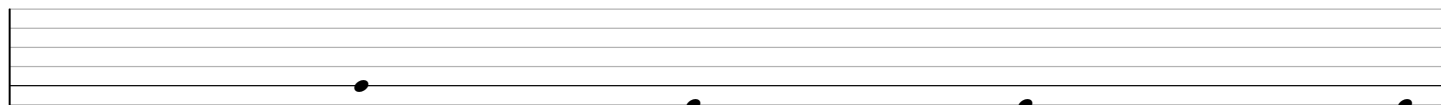


9'00"

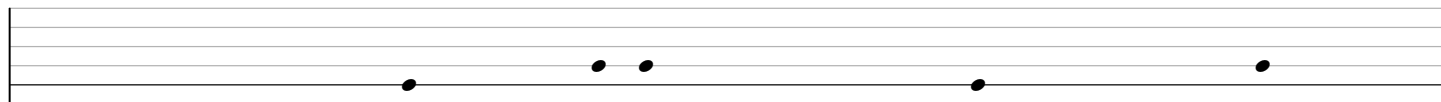
9'04"



9'12"

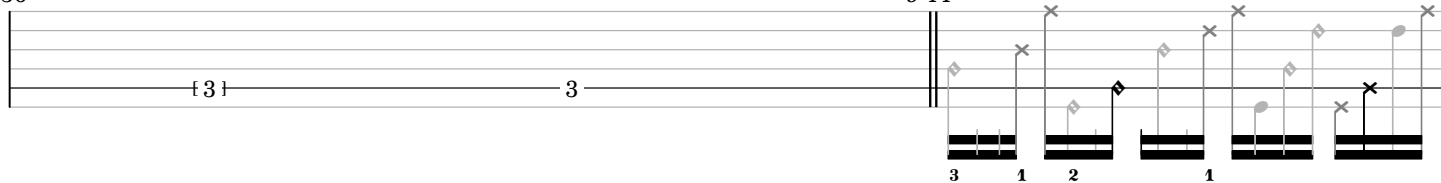


9'24"



9'36"

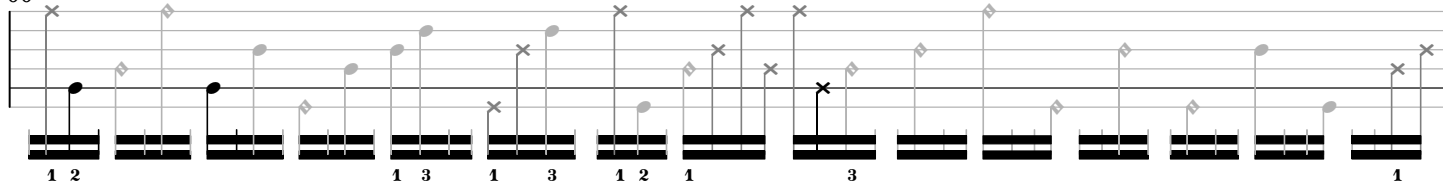
9'44"



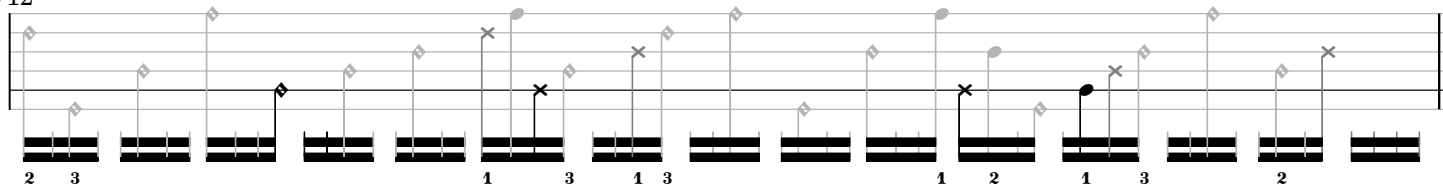
9'48"



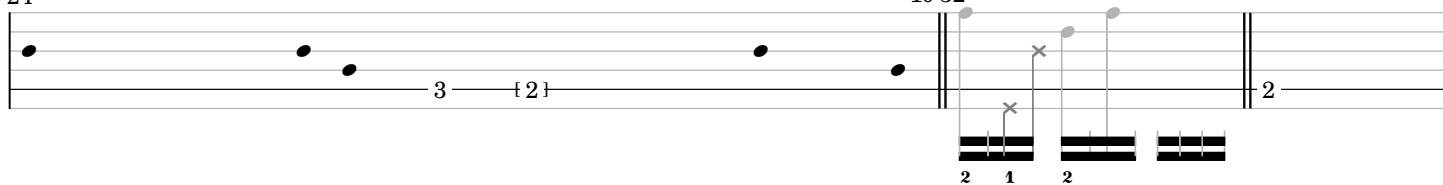
10'00"



10'12"

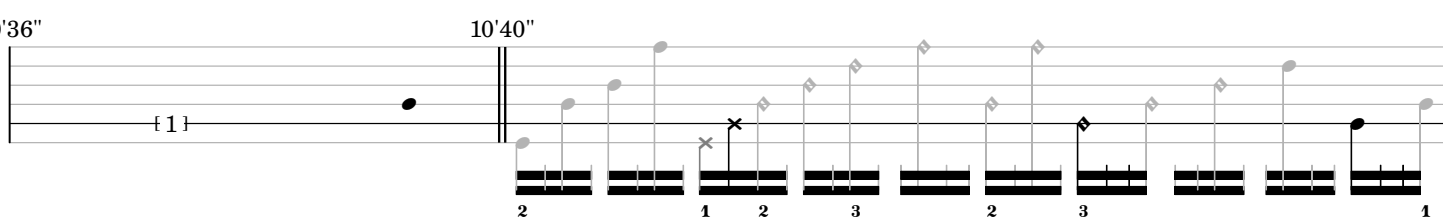


10'24"



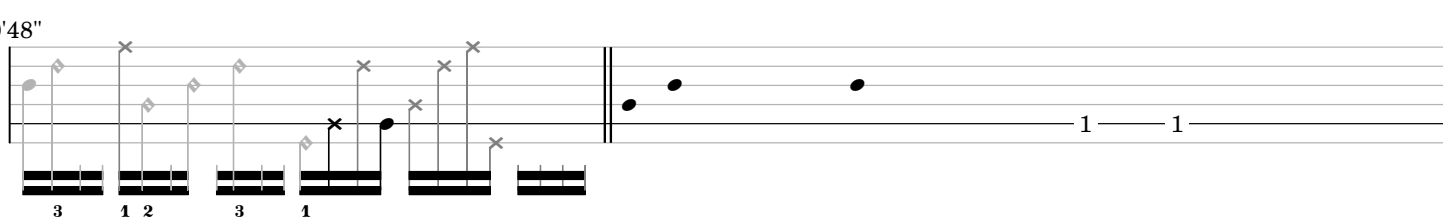
10'32"

10'36"

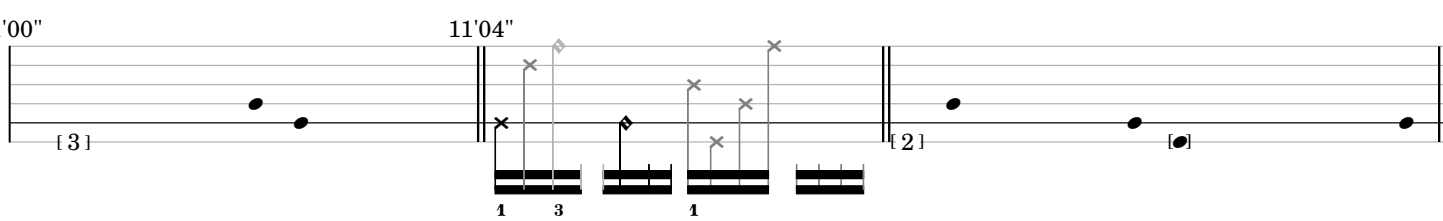


10'40"

10'48"

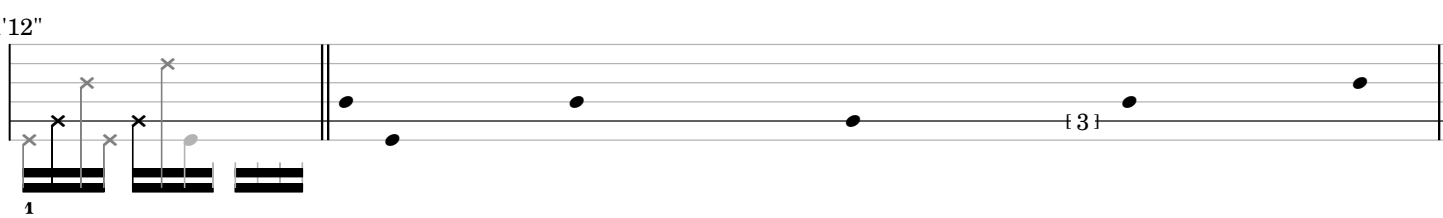


11'00"

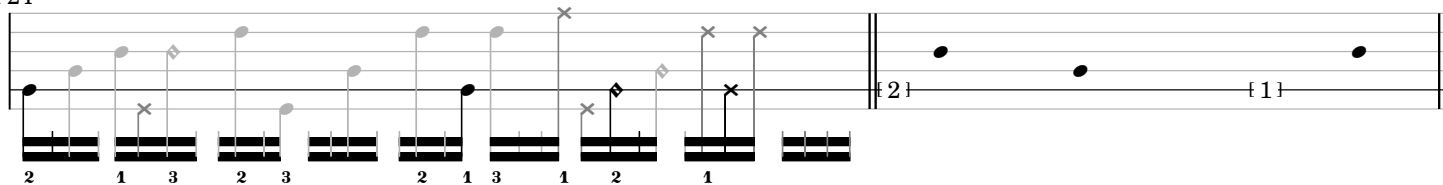


11'04"

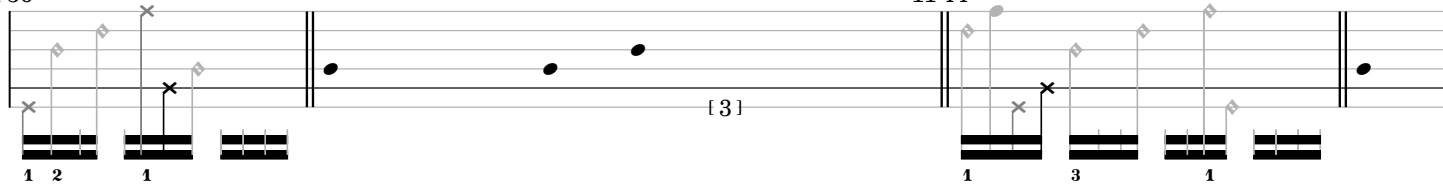
11'12"



11'24"

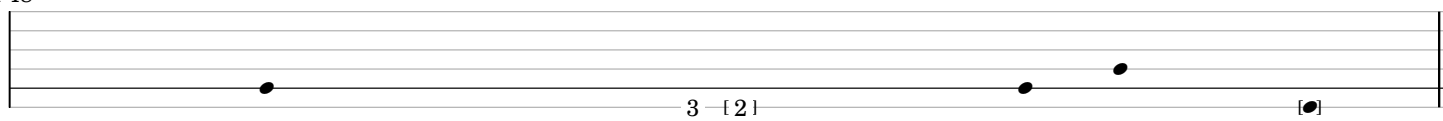


11'36"

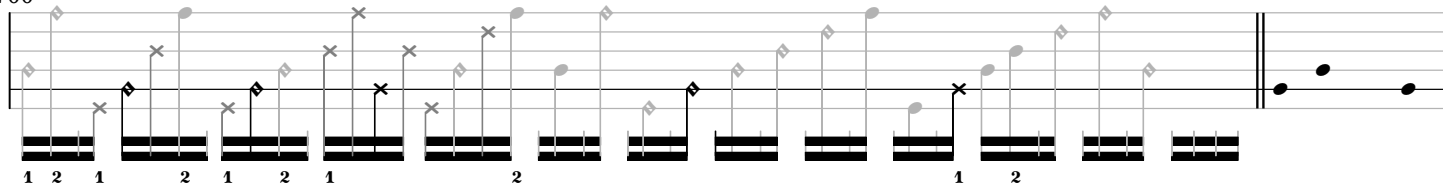


11'44"

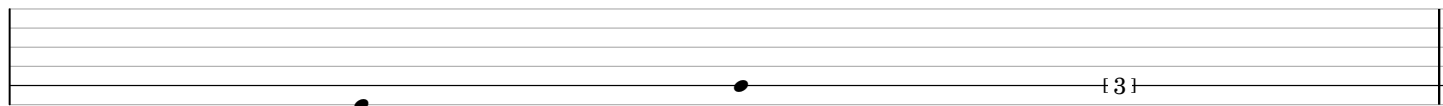
11'48"



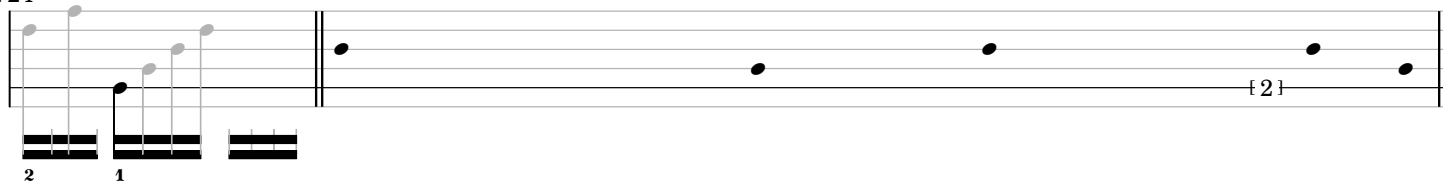
12'00"



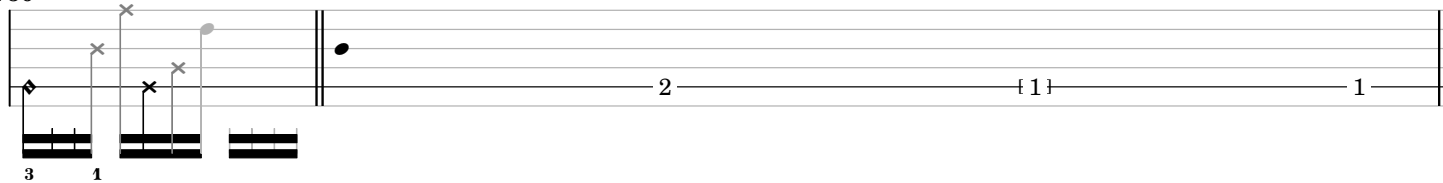
12'12"



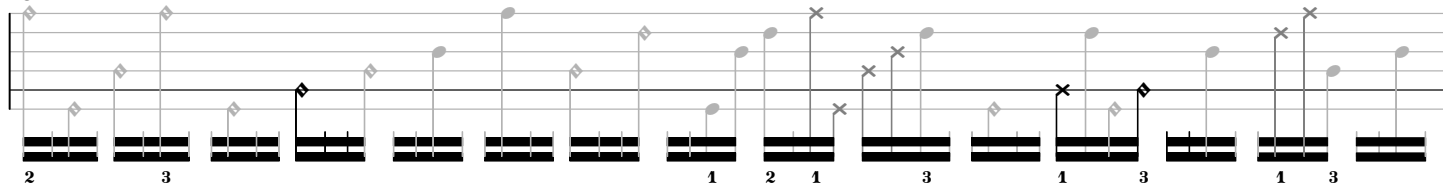
12'24"



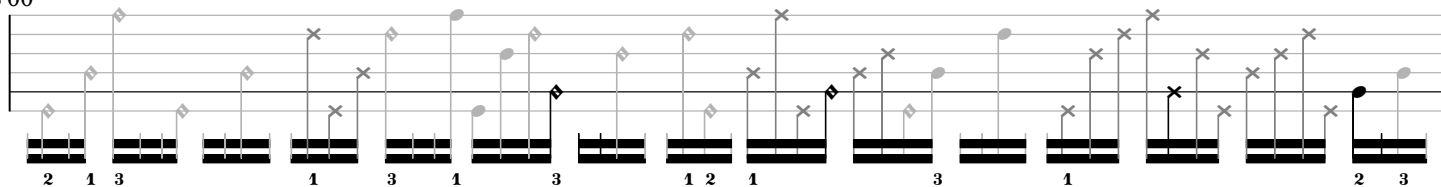
12'36"



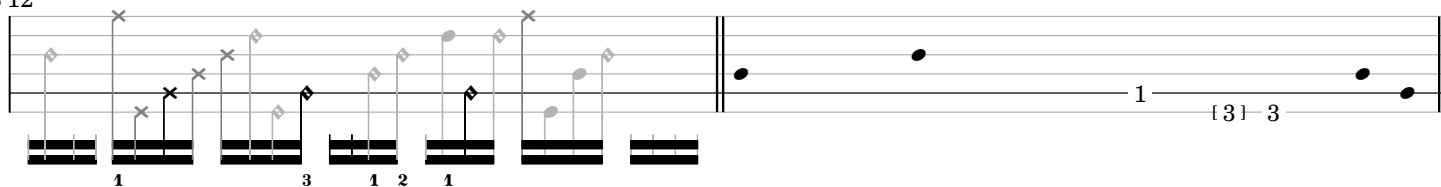
12'48"



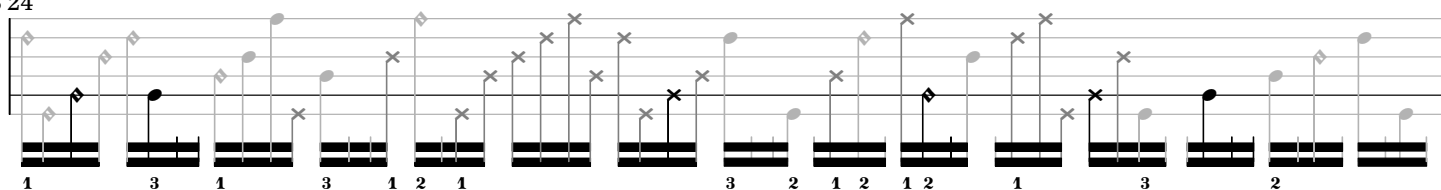
13'00"



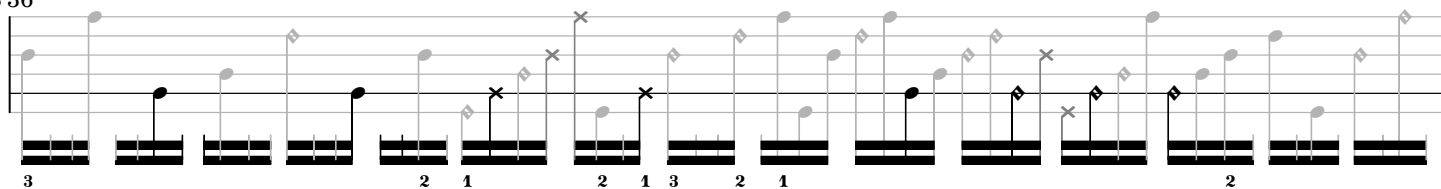
13'12"



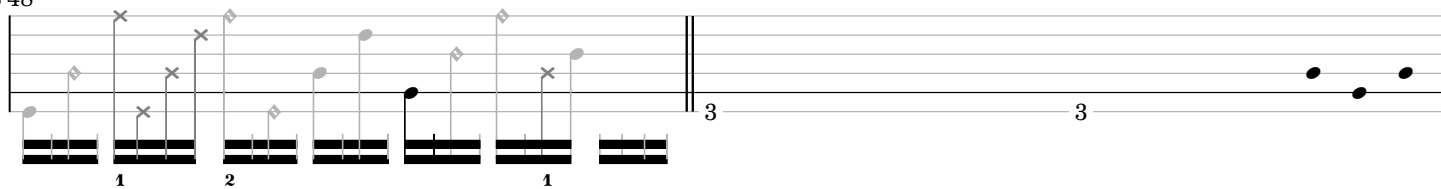
13'24"



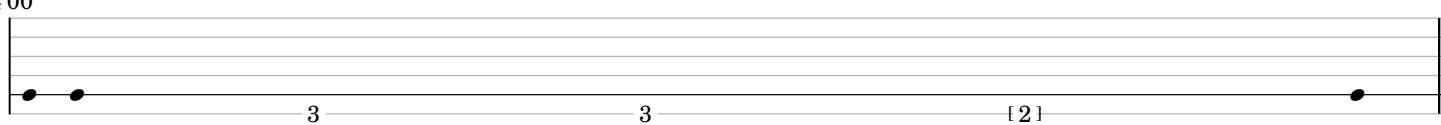
13'36"



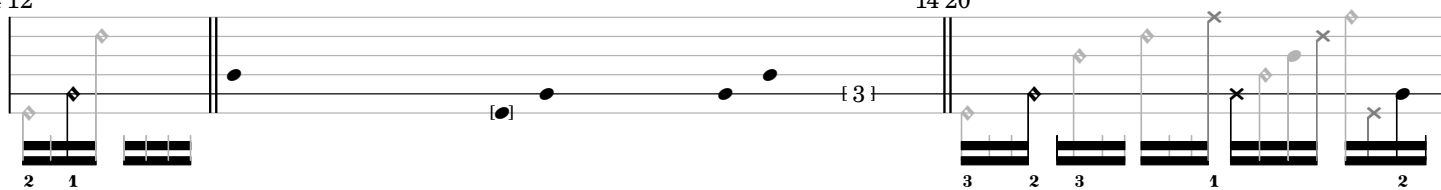
13'48"



14'00"



14'12"



14'20"

14'24"



14'36" 14'44"

3 { 2 }

3 1 2 3 1 2

14'48"

15'00"

15'12"

15'24"

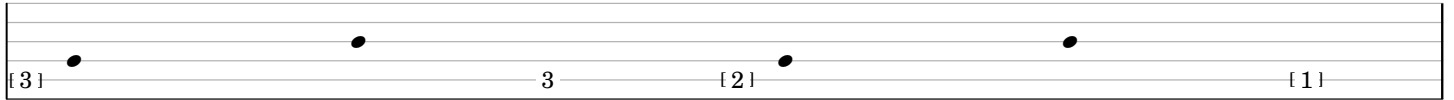
15'36"

ostinato and interrupt

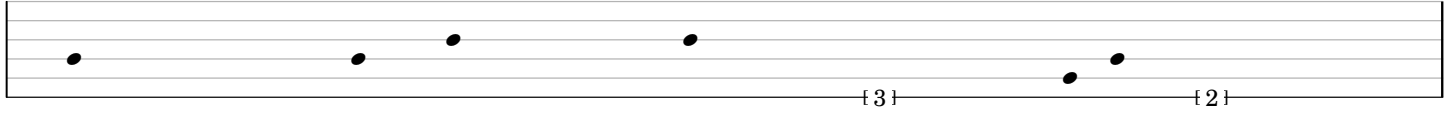
ensemble part 6

michael winter (mexico city, mx; 2017)
version generated: 2017.12.05

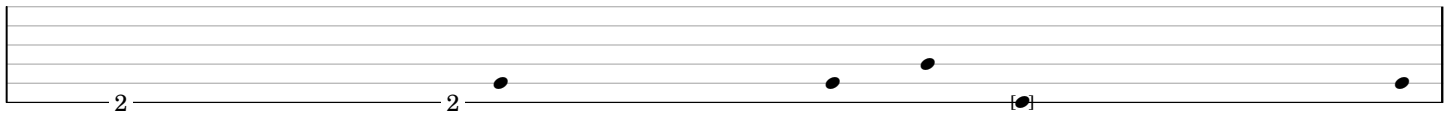
0'12"



0'24"



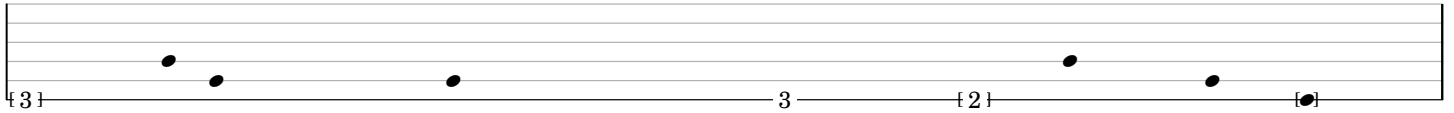
0'36"



0'48"



1'00"

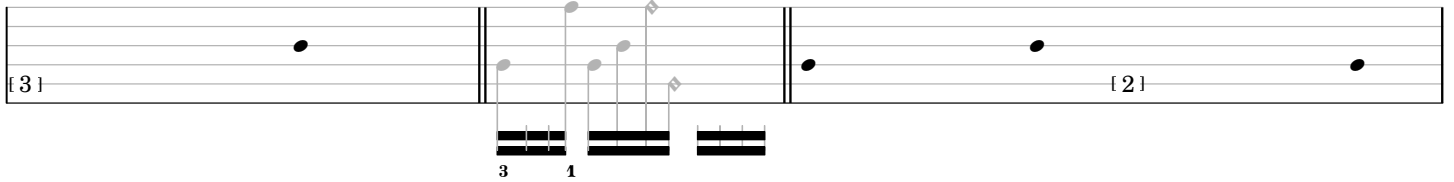


1'12"



1'24"

1'28"



1'36"



1'48"

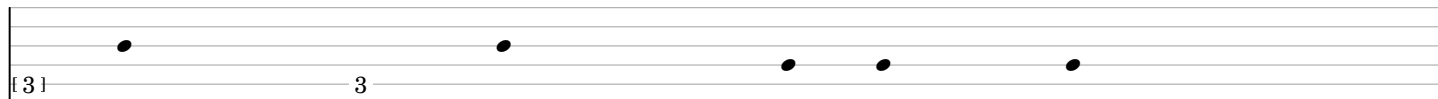


2'00"

2'04"



2'12"

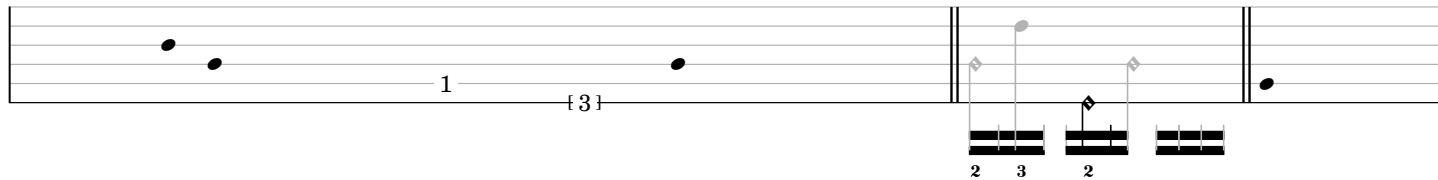


2'24"

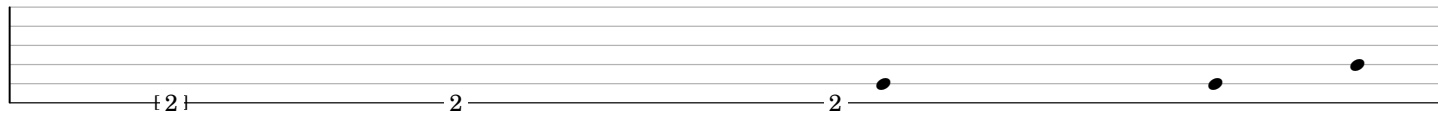


2'36"

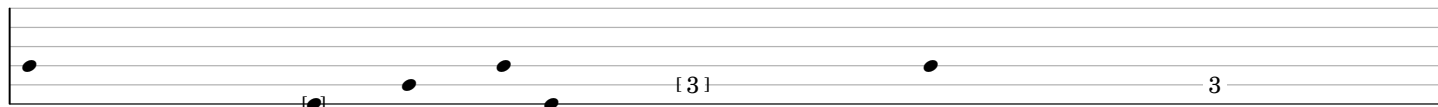
2'44"



2'48"

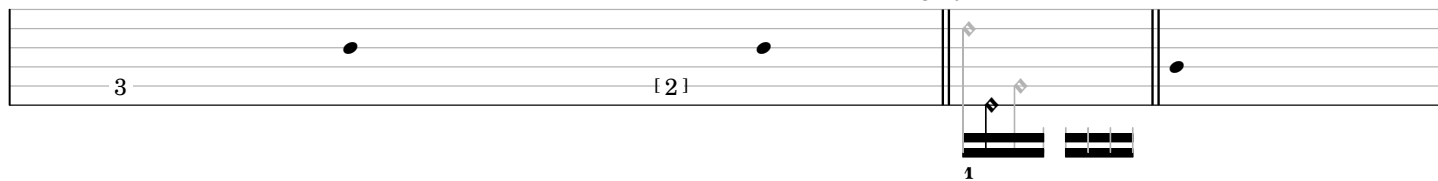


3'00"



3'12"

3'20"



5'00" 5'04"

5'12"

5'24"

5'36" 5'44"

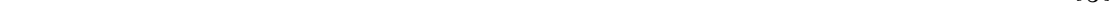
5'48"

6'00"

6'12" 6'16"

6'24"

A number line from 0 to 10. Tick marks are at every integer. Points are plotted at 2, 3, 6, and 9. The points at 2 and 6 are labeled with the fraction $\frac{2}{3}$. The point at 9 is labeled with the fraction $\frac{3}{1}$.

[illegible]

The diagram illustrates the construction of a 3D surface from a 2D grid. The top part shows a 2D grid with points labeled 1, 1, {3}, and {2}. The bottom part shows a 3D surface with points labeled 1, 1, {3}, and {2}.

8'12"

Musical notation for 8'12". The staff shows a sequence of notes with various articulations (diamonds, crosses) and fingerings (3, 1, 3, 1, 3, 1). A double bar line is present. Below the staff, there are two groups of notes with fingerings 3 and 3.

8'24"

Musical notation for 8'24". The staff shows a sequence of notes with fingerings {2} and 2.

8'36"

8'40"

Musical notation for 8'36" and 8'40". The staff shows a sequence of notes with various articulations (diamonds, crosses) and fingerings (1, 2, 1, 3, 1, 2, 1, 2, 1, 3). A double bar line is present.

8'48"

Musical notation for 8'48". The staff shows a sequence of notes with fingerings {3} and 3.

9'00"

9'04"

Musical notation for 9'00" and 9'04". The staff shows a sequence of notes with various articulations (diamonds, crosses) and fingerings (1, 3, 1, 2). A double bar line is present.

9'12"

Musical notation for 9'12". The staff shows a sequence of notes.

9'24"

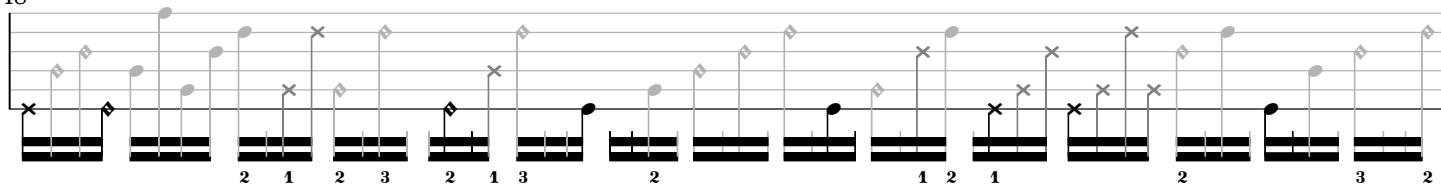
Musical notation for 9'24". The staff shows a sequence of notes.

9'36"

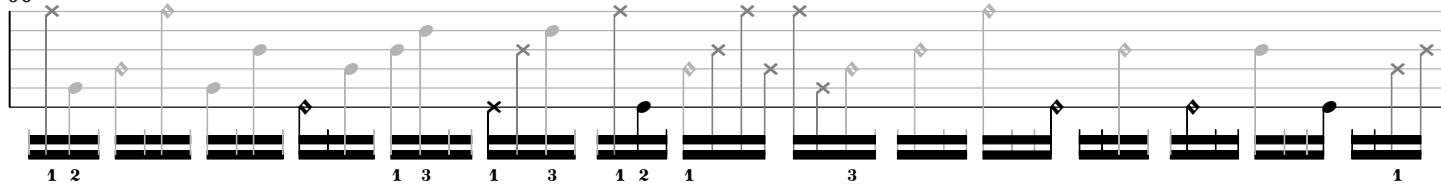
9'44"

Musical notation for 9'36" and 9'44". The staff shows a sequence of notes with various articulations (diamonds, crosses) and fingerings (3, 1, 2, 1). A double bar line is present.

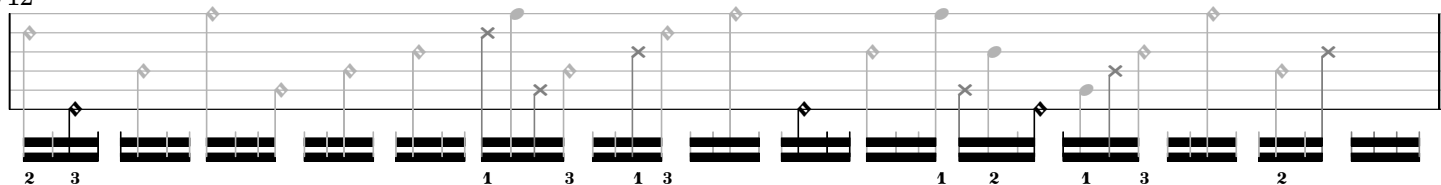
9'48"



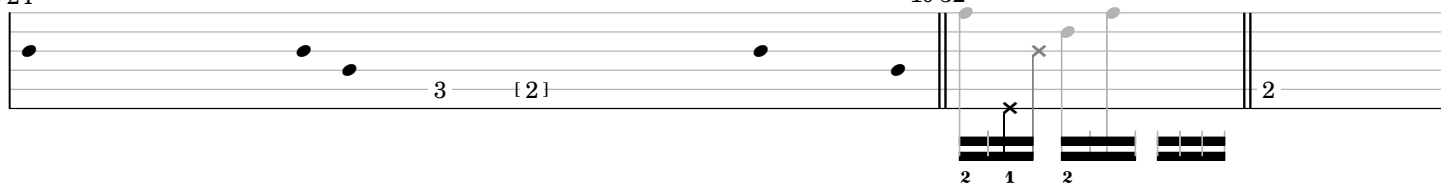
10'00"



10'12"

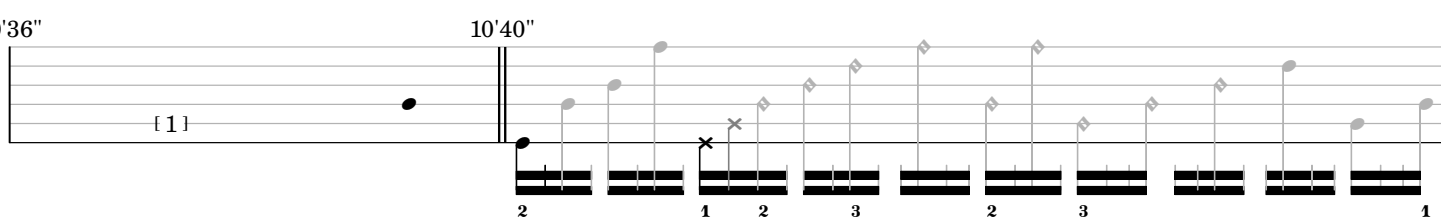


10'24"



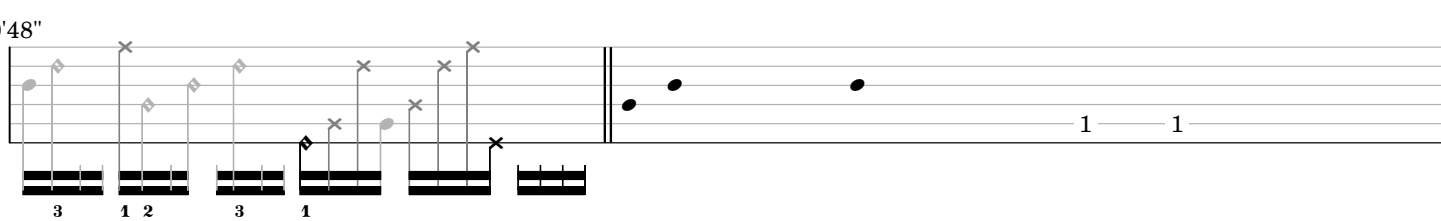
10'32"

10'36"

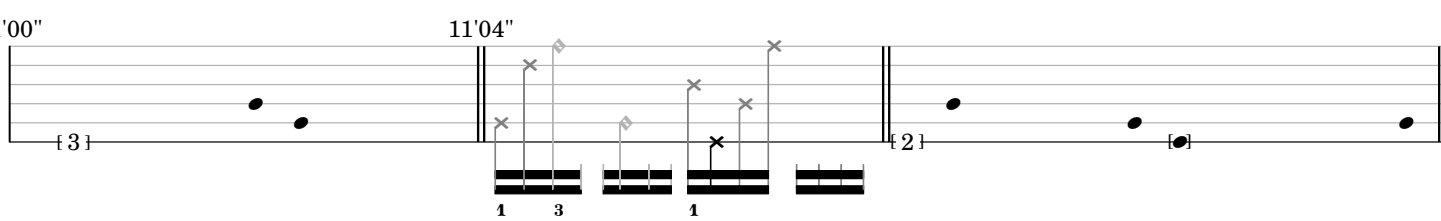


10'40"

10'48"

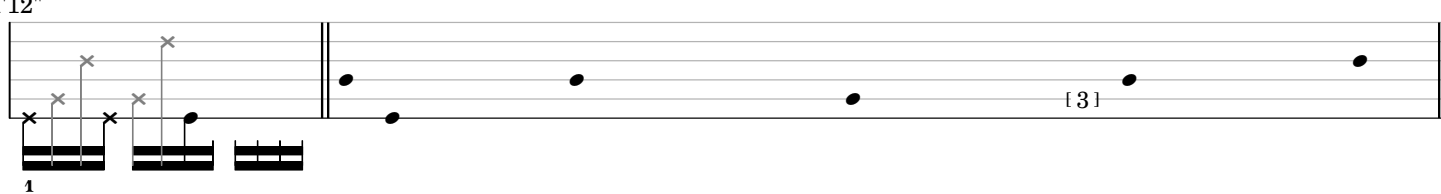


11'00"



11'04"

11'12"



11'24"

Musical notation for 11'24". The staff shows a sequence of notes with various articulations (accents, slurs, and 'x' marks). The bottom staff shows fingerings: 2, 1 3, 2 3, 2 1 3, 1 2, 1. There are repeat signs and bracketed counts: [2] and [1].

11'36"

Musical notation for 11'36". The staff shows a sequence of notes with various articulations. The bottom staff shows fingerings: 1 2, 1. There are repeat signs and bracketed counts: [3].

11'44"

Musical notation for 11'44". The staff shows a sequence of notes with various articulations. The bottom staff shows fingerings: 1, 3, 1. There are repeat signs.

11'48"

Musical notation for 11'48". The staff shows a sequence of notes with various articulations. The bottom staff shows fingerings: 3-[2]. There are repeat signs.

12'00"

Musical notation for 12'00". The staff shows a sequence of notes with various articulations. The bottom staff shows fingerings: 1 2 1, 2, 1 2, 1, 2, 1 2, 2, 1 2, 1 2, 1 2, 1 2. There are repeat signs.

12'12"

Musical notation for 12'12". The staff shows a sequence of notes with various articulations. The bottom staff shows fingerings: [3]. There are repeat signs.

12'24"

Musical notation for 12'24". The staff shows a sequence of notes with various articulations. The bottom staff shows fingerings: 2, 1. There are repeat signs and bracketed counts: [2].

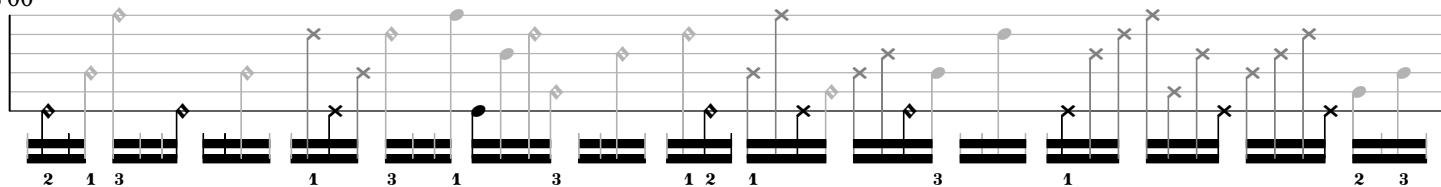
12'36"

Musical notation for 12'36". The staff shows a sequence of notes with various articulations. The bottom staff shows fingerings: 3, 1. There are repeat signs and bracketed counts: [1].

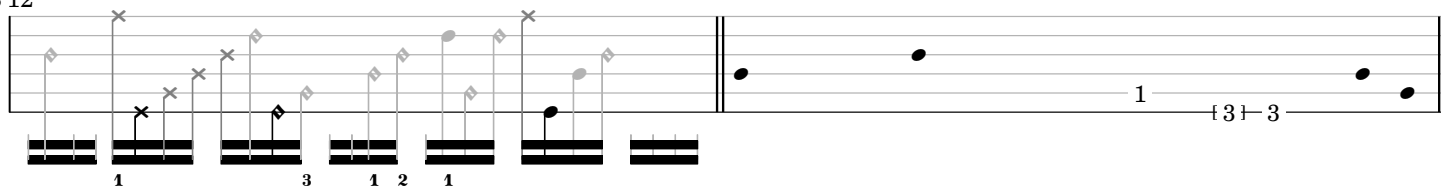
12'48"

Musical notation for 12'48". The staff shows a sequence of notes with various articulations. The bottom staff shows fingerings: 2, 3, 1 2 1, 3, 1 3, 1 3, 1 3. There are repeat signs.

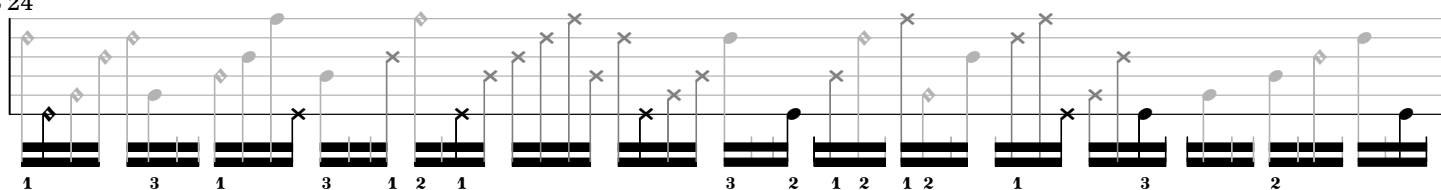
13'00"



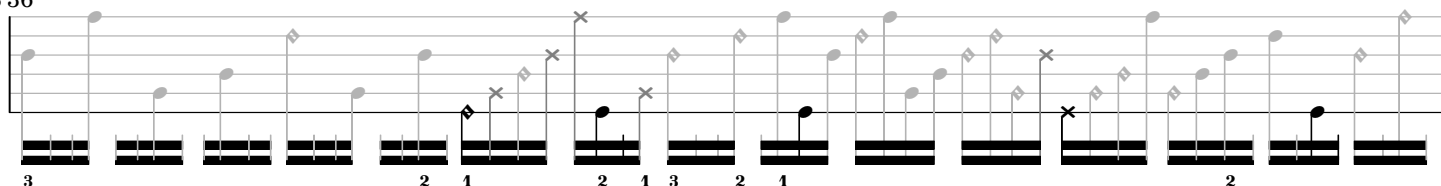
13'12"



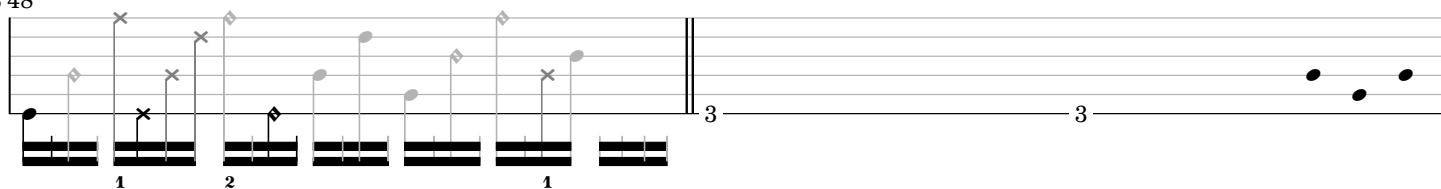
13'24"



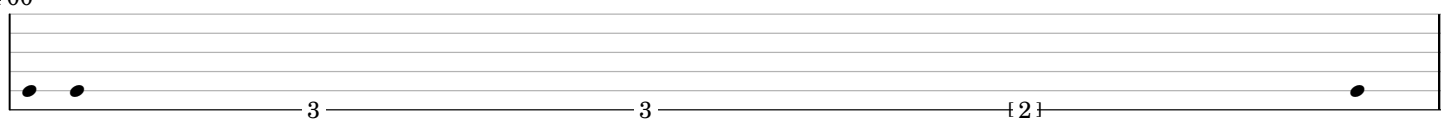
13'36"



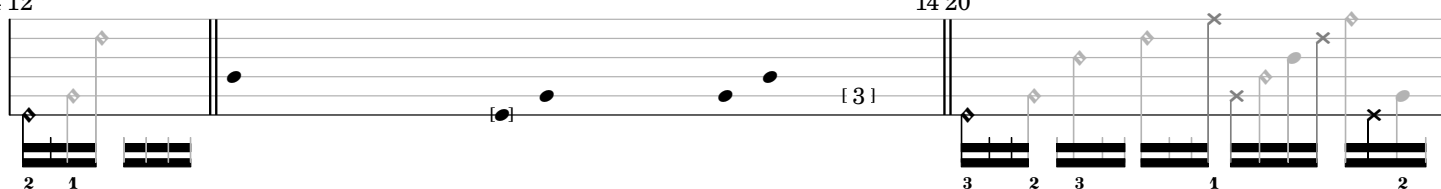
13'48"



14'00"

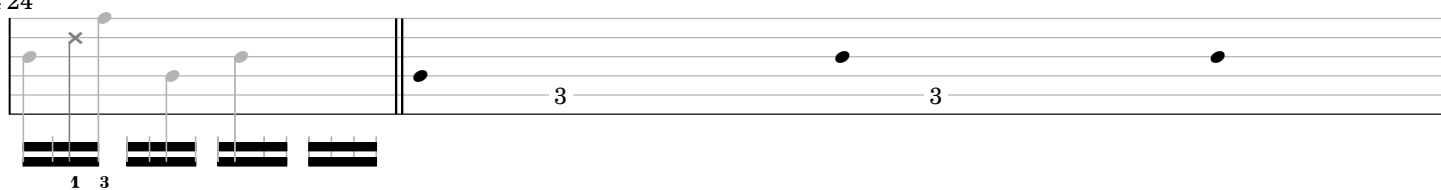


14'12"



14'20"

14'24"



14'36" 14'44"

3 [2] 3 1 2 3 1 2

14'48"

1 2 1 2 3 2 3 1 2 3 2 1

15'00"

3 2 1 2 1 2 3 1 3 2 1 3 2

15'12"

3 1 2 1 3 2 3 2 { 1 } 1

15'24"

2 1 2 1 2 1 3 1 2 1 2 3 { 3 }

15'36"

{ 2 } 2 { 2 }

ostinato_and_interrupt_main.scd

```

1 (
2 // MAIN LAUNCH — Best to reboot interpreter and server first to make sure all buffers are cleared
3 `dir = thisProcess.nowExecutingPath.dirname;
4 "ostinato_and_interrupt_generator.synthdef.scd".loadRelative(true, {
5     "ostinato_and_interrupt_player.synthdef.scd".loadRelative(true, {
6         "ostinato_and_interrupt_nrt_generator_function.scd".loadRelative(true, {
7             "ostinato_and_interrupt_lilypond_generator_function.scd".loadRelative(true, {
8                 "ostinato_and_interrupt_gui_generator_function.scd".loadRelative(true, {
9                     if(File.exists(`dir +/ " ../audio/ostinato_and_interrupt.wav"), {
10                         `appStatusString = "loading buffer";
11                         `generateGUI.value;
12                         `appStatusFunc.play;
13                         Buffer.read(s, `dir +/ " ../audio/ostinato_and_interrupt.wav", action:
14                             {
15                                 |buf| `totalDur = buf.duration;
16                                 `play = Synth.new(\play, [\buf, buf]);
17                                 `appStatusFunc.stop;
18                                 {"appStatus.string = "ready"}.defer
19                             });
20                     }, {
21                         `appStatusString = "generating data";
22                         `generateGUI.value;
23                         `appStatusFunc.play;
24                         `generateData.value(true));
25 }}}}}});

```

ostinato_and_interrupt_generator.synthdef.scd

```

1 (
2 //~~SYNTHDEF THAT GENERATES THE PIECE
3 `ostinato_and_interrupt = SynthDef(\ostinato_and_interrupt, {
4
5     // TODO: Replace PlayBuf with PlayBufCF
6
7     //-----ARG DECLARATION-----
8     arg scoreBuf, seed = 20170121, rt = 0, // set score buffer, random seed, and if run in realtime (sets outputs
9         differently)
10     percBufDelims = #[0, 1, 2, 3, 4, 5, 6], // set buffers for percussion samples
11     guitOpenBufDelims = #[0, 1, 2, 3, 4, 5, 6], // set buffers for open strings
12     guitHarmBufDelims = #[0, 1, 2, 3, 4, 5, 6], // set buffers for harmonics
13     guitMuteBufDelims = #[0, 1, 2, 3, 4, 5, 6], // set buffers for muted strings
14     guitBassBufDelims = #[0, 1, 2, 3, 4, 5, 6]; // set buffers for bass notes
15
16     //-----VAR DECLARATION-----
17     var fund, durUnit, tuning, oBassNotes, oOpenStringNotes; // static vars
18     var li, trig, lastoString, lastoDurTemp, switchTrig, lastoStepper; // feedback in
19     var env, iEnv, oEnv, currentDur, switch; // switch
20     var oTrig, oStepper, oBassNote, oString, oNote, oFreq, oDurTemp, oDur, oStringCounts,
21         oPluckedStrings, oSwitchTrig, oSwitchArm, oSwitchTrigDel; //ostinato
22     var startState, endState, endTrig; // start and end states
23     var iTrig, iSound, iDur, iString, iEnsemble, iPerc, iPluckedStrings,
24         iSwitchTrig, iSwitchArm, iSwitchTrigDel; // interrupt
25     var aDust, aLBuf, aFlicker, aBNoise, aWNoise, aSinBeat, aSinTrig, aFadeIn, aFadeOut; // electronic
26         accompaniment
27     var count; // score
28     RandSeed.ir(1, seed); // random seed — change for different results
29
30     //-----STATIC VARS-----
31     fund = 38; // fundamental in midi (guitar low d)
32     durUnit = 0.2; // minimum durational unit
33     tuning = [1/1, 3/2, 2/1, 5/2, 7/2, 4/1]; // tuning of the open strings
34     oBassNotes = [10, 9, 8, 3, 2, 0]; // bass notes relative to fundamental in semitones
35     oOpenStringNotes = [0, 7, 12, 16]; // open string notes relative to fundamental in semitones (used only for
36         score)
37
38     //-----FEEDBACKIN-----
39     li = LocalIn.kr(6, [0, -1, -1, 0, 0, 1]); // init vals
40     trig = li[0] + TDelay.kr(Impulse.kr(0), 12); // start the guitar 12 seconds after electronic accompaniment
41     lastoString = li[1]; // get last ostinato string for
42     lastoDurTemp = li[2]; // get last duration value (not yet multiplied by durUnit)
43     lastoStepper = li[3]; // last position of the ostinato
44     switchTrig = li[4]; // trigger to change from ostinato to interrupt
45     startState = li[5]; // delay before interrupts can start to occur
46
47     //-----SWITCH-----
48     env = EnvGen.kr(Env.new([0, 1], 60 * 13, \sine), 1 - startState); // env that controls the switching between
49         ostinato and interrupt
50     oEnv = 0.25 + (env * 3.75); // chance that ostinato will switch to interrupt over time
51     iEnv = 1 + (env * 29); // chance that interrupt will switch to ostinato over time
52     currentDur = (Line.kr(0, 60 * 20, 60 * 20) / 0.2).trunc; // time tracker
53     switch = Stepper.kr(switchTrig, 0, 0, 1); // state — ostinato or interrupt
54     Poll.kr(trig, currentDur, \currentDur); // poll current duration
55     Poll.kr(trig, env, \env); // monitor current position of envelope
56
57     //-----OSTINATO-----
58     // mute triggers when off
59     oTrig = trig * (switch <= 0);
60     // update string counts
61     oStringCounts = { arg i;
62         var isString = (i <= lastoString) * (lastoString <= i);

```

```

59     PulseCount.kr(oTrig * isString, Changed.kr(lastoStepper)) } ! 4;
60 // step through the bass note of the ostinato once all notes in a cell have been played
61 oStepper = Stepper.kr((oTrig * TWChoose.kr(oTrig, [0, 1], [1, 1], 1) * (Mix.new(oStringCounts > 0) >= 3)), 0,
62     0, 5);
63 oBassNote = Select.kr(oStepper, oBassNotes);
64 // select duration favoring a change for longer notes to promote flurries of shorter notes
65 oDurTemp = TWChoose.kr(oTrig * TWChoose.kr(oTrig, [0, 1], [(((lastoDurTemp <= 0) * 0.75) + 1) * (lastoDurTemp >
66     5), 1], 1),
67     [1, 2, 3, 4, 5, 6, 7], Select.kr(oStepper < 3, [[4, 3, 2, 2, 2, 1, 1], [2, 3, 3, 3, 2, 1, 1]], 1);
68 // add jitter to duration (if this is odd unfortunately a rounding error is produced)
69 oDur = 2 * (oDurTemp + TIRand.kr(0, Select.kr(oDurTemp < 3, [2, 0], oTrig));
70 // select string (always oBassNote if oBassNote has been stepped to next) and promote change on shorter notes
71 oString = TWChoose.kr(oTrig,
72     Select.kr(oStepper < 3, [[0, 1, 2], [1, 2, 3]]),
73     Select.kr(Changed.kr(oStepper), [[4, 3, 3] * (1 / pow(Select.kr(oStepper < 3,
74         [oStringCounts.drop(-1), oStringCounts.drop(1)]) + 1, Select.kr(lastoDurTemp < 2, [0.75, 2]))],
75         [1, 0, 0]), 1);
76 // select note based on string
77 oNote = Select.kr((oString - (oStepper < 3)) > 0, [oBassNote, Select.kr(oString, oOpenStringNotes)]);
78 // play guitar
79 oPluckedStrings = { |i| var string, isString, isStringDel, freq, snd;
80     string = 5 - i; // invert string number (since oString 1 is guitar string IV)
81     isString = oTrig * (string <= (5 - oString)) * ((5 - oString) <= string); // check if string is
82     triggered
83     isStringDel = TDelay.kr(isString, 0.01); // slight delay for envelope
84     // play samples and select open string or bass note
85     snd = PlayBuf.ar(1, Select.kr(Latch.kr((oString - (oStepper < 3)) > 0, isStringDel),
86         [TIRand.kr(Select.kr(oStepper, guitBassBufDelims), Select.kr(oStepper, guitBassBufDelims) - 1,
87             isStringDel),
88             TIRand.kr(guitOpenBufDelims[string], guitOpenBufDelims[string + 1] - 1, isStringDel)
89         ]), Latch.kr(1, isStringDel), isStringDel) * (1 - EnvGen.ar(Env.new([0, 1, 0], [0.01, 0.01]), isString
90         ));
91     Out.ar(Select.kr(rt, [string - 2, [0, 1]]), snd * (1/3)); // ostinato guitar records to channels 0 - 3
92 } ! 4;
93 /* karplus strong version - legacy sonification used for auditioning the piece
94 oFreq = Select.kr((oString - (oStepper < 3)) > 0, [(oNote + fund).midicps, fund.midicps * Select.kr(oString,
95     tuning)]);
96 oPluckedStrings = { arg i;
97     var string, isString, snd;
98     string = 5 - i;
99     isString = oTrig * (string <= oString) * (oString <= string);
100     snd = Pluck.ar(WhiteNoise.ar(0.1), isString, 0.2,
101         Latch.kr(oFreq.reciprocal, isString), 10, 0);
102     Out.ar(Select.kr(rt, [string, [0, 1]]), snd);
103 } ! 4;*/
104 // endState - envelope has finished but cycles through switches until the final bass note is played
105 endState = Latch.kr(1, trig * (env >= 1) * (switch > 0) * (oStepper <= 2) * (oStepper >= 2));
106 endTrig = trig * endState * (oStepper >= 5);
107 FreeSelf.kr(TDelay.kr(endTrig, (oDur * durUnit) + 10));
108 // trigger switch and allow notes to pass until next 5 second interval (so interrupt always starts on a common
109 // multiple of 4 and 5)
110 oSwitchTrig = (startState <= 0) * (endState <= 0) * oTrig * (PulseCount.kr(Changed.kr(oStepper), Changed.kr(
111     switch)) > (4 - oEnv)) *
112     TWChoose.kr(oTrig, [0, 1], [1 - (env >= 1), oEnv], 1);
113 oSwitchArm = PulseCount.kr(oSwitchTrig, switchTrig); // switch trig armed
114 oSwitchTrigDel = 20 - (currentDur % 20); // count down to switch
115 oDur = Select.kr(oSwitchArm <= 0, [Clip.kr(oDur, 0, oSwitchTrigDel), oDur]); // clip dur if past switch
116 oSwitchTrig = oTrig * (oDur >= oSwitchTrigDel) * (oSwitchArm > 0); // trigger the switch
117 Poll.kr(oTrig, oDur, \oDur); // monitor duration of the ostinato note
118
119 //-----Interrupt-----
120 // mute triggers when off
121 iTrig = trig * switch;
122 // select sound type: mute, harmonic, or open string (make first interruption not have percussion)
123 iSound = TWChoose.kr(iTrig * TWChoose.kr(iTrig, [0, 1], [1, 2], 1), [0, 1, 2],
124     Select.kr(PulseCount.kr(Changed.kr(switch)) > 1, [[1, 1, 0], [1, 1, 1]], 1);
125 // select duration
126 iDur = Select.kr(iSound < 2,
127     [1, TChoose.kr(iTrig * TWChoose.kr(iTrig, [0, 1], [1, 1], 1), [1, 2, 3])]);
128 // select string as a stepper occasionally changing how many strings are skipped
129 iString = Stepper.kr(iTrig, 0, 5, TWChoose.kr(iTrig, [1, 2, 3], [3, 2, 1], 1));
130 // play ensemble
131 iEnsemble = { |i| var string, isString, harm, freq, rel, fade, snd;
132     string = 5 - i; // invert string number (since iString 1 is guitar string IV)
133     isString = iTrig * (string <= (5 - iString)) * ((5 - iString) <= string); // check if string is
134     triggered
135     harm = TRand.kr(1 + iSound, iSound * 5 + 1, isString); // set based on open string or harmonic
136     freq = fund.midicps * tuning[i] * harm; // calculate freq
137     rel = Select.kr(switch, [2, 0]); // set release
138     fade = EnvGen.kr(Env.asr(releaseTime: rel), // gate if sound is mute or with switch
139         Latch.kr(Select.kr(iSound < 2 * switch, [0, 1]), isString + (Changed.kr(switch) * (1 - switch)
140             ));
141     snd = SinOsc.ar(Latch.kr(freq, isString), 0, fade * (1/harm)); // simple sine tone with amp 1/harmonic
142     number
143     Out.ar(Select.kr(rt, [string + 16, [0, 1]]), snd * (1/12)); // interrupt ensemble records to channels
144     16 - 21
145 } ! 6;
146 iPerc = { |i| var string, isString, isStringDel, freq, snd;
147     string = 5 - i; // invert string number (since iString 1 is guitar string IV)
148     isString = iTrig * (string <= (5 - iString)) * ((5 - iString) <= string) * (iSound >= 2); // check if
149     sound is percussion / mute
150     isStringDel = TDelay.kr(isString, 0.01); // slight delay for envelope
151     // play samples
152     snd = PlayBuf.ar(1, TIRand.kr(percBufDelims[string], percBufDelims[string + 1] - 1, isStringDel),

```

```

139         Latch.kr(1, isStringDel), isStringDel) * (1 - EnvGen.ar(Env.new([0, 1, 0], [0.01, 0.01])),
140         isString));
141     Out.ar(Select.kr(rt, [string + 10, [0, 1]]), snd * (1/2)); // interrupt ensemble records to channels 10
142     - 15
143 } ! 6;
144 iPluckedStrings = { |i| var string, isString, isStringDel, snd;
145     string = 5 - i; //invert string number (since iString 1 is guitar string IV)
146     isString = iTrig * (string <= (5 - iString)) * ((5 - iString) <= string); // check if string is
147     triggered
148     isStringDel = TDelay.kr(isString, 0.01); // slight delay for envelope
149     // play samples and select open string, harmonic, or muted string
150     snd = PlayBuf.ar(1, Select.kr(Latch.kr(iSound, isStringDel),
151     [TIRand.kr(guitOpenBufDelims[string], guitOpenBufDelims[string + 1] - 1, isStringDel),
152     TIRand.kr(guitHarmBufDelims[string], guitHarmBufDelims[string + 1] - 1, isStringDel),
153     TIRand.kr(guitMuteBufDelims[string], guitMuteBufDelims[string + 1] - 1, isStringDel)
154     ]), Latch.kr(1, isStringDel), isStringDel) * (1 - EnvGen.ar(Env.new([0, 1, 0], [0.01, 0.01])), isString)
155     );
156     Out.ar(Select.kr(rt, [string + 4, [0, 1]]), snd * (1/3)); // interrupt ensemble records to channels 4 -
157     9
158 } ! 6;
159 /* karplus strong version - legacy sonification used for auditioning the piece
160 iPluckedStrings =
161 { |i| var string, isString, freq, snd;
162     string = 5 - i;
163     isString = iTrig * (i <= iString) * (iString <= i) * (iSound < 2);
164     freq = fund.midicps * tuning[i] *
165     TRand.kr(1 + iSound, iSound * 5 + 1, isString);
166     snd = Pluck.ar(WhiteNoise.ar(0.1), isString, 0.2,
167     Latch.kr(freq.reciprocal, isString), 10, 0);
168     Out.ar(Select.kr(rt, [string + 4, [0, 1]]), snd);
169 } ! 6;*/
170 // trigger switch and allow notes to pass until next interval of 4 pulses then add 4 pulses (5 pulses = 1
171 second)
172 iSwitchTrig = iTrig * TWChoose.kr(iTrig, [0, 1], [iEnv, 1], 1);
173 iSwitchArm = PulseCount.kr(iSwitchTrig, switchTrig); // switch trig armed
174 iSwitchTrigDel = 4 - (currentDur % 4); // countdown to switch
175 iDur = Select.kr(iSwitchArm <= 0, [Clip.kr(iDur, 0, iSwitchTrigDel), iDur]); // clip dur if past switch
176 iSwitchTrig = iTrig * (iDur >= iSwitchTrigDel) * (iSwitchArm > 0); // trigger the switch
177 iDur = Select.kr(iSwitchTrig, [iDur, iDur + 4]); // add 4 pulses
178 Poll.kr(iTrig, iDur, \iDur); // monitor duration of interrupt note
179
180 //-----Electronic Accompaniment-----
181 aDust = Dust.kr(10); // random triggers
182 aLBuf = LocalBuf.new((SampleRate.ir / fund.midicps).trunc); // buffer for harmonic flickering sound
183 // fill buf with random bursts of noise
184 RecordBuf.ar(PinkNoise.ar(), aLBuf, run: Latch.ar(TWChoose.kr(aDust, [0, 1], [5, 1], 1), aDust + TDelay.kr(
185     aDust, 0.01));
186 // play buf back at a rate equivalent to the bass note in the ostinato, select whether on or off on every
187 ostinato note
188 aFlicker = PlayBuf.ar(1, aLBuf, 1, Impulse.ar((fund + oBassNote).midicps), loop: 1) * Latch.ar(
189     TWChoose.kr(aDust, [0, 1], [1, 2], 1), aDust) * (1 - switch) * TWChoose.kr(oTrig, [0, 1], [2, 1], 1) *
190     0.02;
191 aBNoise = BrownNoise.ar(0.007) * (1 - switch); // brown noise during ostinato
192 aWNoise = WhiteNoise.ar(0.002) * switch; // white noise during interrupt
193 // sine tone beating that can change rate (or not) on every ostinato note (around fundamental for ostinato and
194 a fifth down for interrupt
195 aSinTrig = TWChoose.kr(oTrig, [0, 1], [1, 1], 1) + Changed.kr(switch);
196 aSinBeat = (SinOsc.ar((fund - (switch * 5)).midicps) +
197     SinOsc.ar((fund - (switch * 5)).midicps + 0.5 + Latch.ar(TRand.kr(0, 2.5, aSinTrig), aSinTrig))) *
198     0.05;
199 aFadeIn = EnvGen.kr(Env.new([0, 1], [1])); // short fade at start of piece
200 aFadeOut = EnvGen.kr(Env.cutoff(10), TDelay.kr(endTrig, oDur * durUnit)); // longer fade at end of piece
201 Out.ar(Select.kr(rt, [22, [0, 1]]), aBNoise * aFadeIn * aFadeOut); // brown noise records to channel 22
202 Out.ar(Select.kr(rt, [23, [0, 1]]), aWNoise * aFadeIn * aFadeOut); // white noise records to channel 23
203 Out.ar(Select.kr(rt, [24, [0, 1]]), aSinBeat * aFadeIn * aFadeOut); // sine beating records to channel 24
204 Out.ar(Select.kr(rt, [25, [0, 1]]), aFlicker * aFadeIn * aFadeOut); // harmonic flickering records to channel
205     25
206
207 //-----Score-----
208 // write score buffer for lilypond transcription
209 count = PulseCount.kr(trig);
210 BufWr.kr(Select.kr(switch,
211     [[switch, oString, oNote, 0, oDur], [switch, iString, 0, iSound, iDur]]), scoreBuf, Select.kr(trig,
212     [-1, count]));
213 Poll.kr(trig, count, \scoreCount);
214
215 //-----FeedbackOut-----
216 LocalOut.kr([
217     // feedback note trigger
218     TDelay.kr(trig * (1 - endTrig), Select.kr(switch, [oDur, iDur]) * durUnit - ControlRate.ir.reciprocal),
219     // feedback oString, oDurTemp, and oStepper for counts
220     oString,
221     oDurTemp - 1,
222     oStepper,
223     // feedback switchTrig
224     TDelay.kr(Select.kr(switch, [oSwitchTrig, iSwitchTrig]), Select.kr(switch, [oDur, iDur]) * durUnit -
225     ControlRate.ir.reciprocal),
226     // feedback whether or not in start state
227     PulseCount.kr(Changed.kr(oStepper)) <= 10
228 ]);
229 });
230 )
231

```

```

219 /*
220 (
221 // Uncomment to test / play everything in real time - see `generateData` for comments of synth messages
222 var sample.index = 0, seed = 20170121;
223 var perc.allocs, quit.open.allocs, quit.harm.allocs, quit.mute.allocs, quit.bass.allocs;
224 var allocMSGs = {
225   arg inFolder;
226   var bufAllocs = [], bufDelims = [];
227   PathName(inFolder).folders.sort({arg a, b; a.folderName[0..1].asInteger < b.folderName[0..1].asInteger }).
     collect({
228     |folder|
229     bufDelims = bufDelims.add(sample.index);
230     folder.files.collect({|file|
231       bufAllocs = bufAllocs.add([\b.allocRead, sample.index, file.fullPath]);
232       sample.index = sample.index+1;
233     });
234   });
235   bufDelims = bufDelims.add(sample.index);
236   [bufAllocs, bufDelims]
237 };
238
239 `dir = thisProcess.nowExecutingPath.dirname;
240 perc.allocs = allocMSGs.value(`dir +/ " ../samples/percussion/");
241 quit.open.allocs = allocMSGs.value(`dir +/ " ../samples/strings.open/");
242 quit.harm.allocs = allocMSGs.value(`dir +/ " ../samples/strings.harmonics/");
243 quit.mute.allocs = allocMSGs.value(`dir +/ " ../samples/strings.muted/");
244 quit.bass.allocs = allocMSGs.value(`dir +/ " ../samples/ostinato.bass/");
245
246 s.listSendBundle(0,
247   perc.allocs[0] ++ quit.open.allocs[0] ++ quit.harm.allocs[0] ++ quit.mute.allocs[0] ++ quit.bass.allocs[0] ++
248   [[\b.alloc, sample.index + 1, 10000, 5], [\d.recv, `ostinato.and.interrupt.asBytes(s);]]);
249
250
251 Synth.new(\ostinato.and.interrupt, [\scoreBuf, sample.index + 1, \seed, seed, \rt, 1,
252   \percBufDelims, perc.allocs[1],
253   \quitOpenBufDelims, quit.open.allocs[1],
254   \quitHarmBufDelims, quit.harm.allocs[1],
255   \quitMuteBufDelims, quit.mute.allocs[1],
256   \quitBassBufDelims, quit.bass.allocs[1]
257 ]);
258 )
259 */

```

ostinato.and.interrupt_nrt_generator_function.scd

```

1 (
2 //~~FUNCTION THAT GENERATES THE PIECE (calls SynthDef(\ostinato.and.interrupt))
3 `generateData = {
4   arg isLaunch = false, seed = 20170121; // set if ran on application launch and random seed
5   var sample.index = 0; // init sample index
6   var perc.allocs, quit.open.allocs, quit.harm.allocs, quit.mute.allocs, quit.bass.allocs; // allocation messages
7
8   // this function reads all the subfolders creating allocation messages and an array that tells synth the range
9   // of each type of sample
10  var allocMSGs = {
11    arg inFolder;
12    var bufAllocs = [];
13    var bufDelims = [];
14    PathName(inFolder).folders.sort({arg a, b; a.folderName[0..1].asInteger < b.folderName[0..1].asInteger
15      }).collect({
16        |folder|
17        bufDelims = bufDelims.add(sample.index);
18        folder.files.collect({|file|
19          bufAllocs = bufAllocs.add([0, [\b.allocRead, sample.index, file.fullPath]]); //
20            allocation message
21            sample.index = sample.index+1;
22        });
23      });
24    bufDelims = bufDelims.add(sample.index); // start and end buffer index for each subfolder
25    [bufAllocs, bufDelims]
26  };
27
28  // run the above function on each folder
29  perc.allocs = allocMSGs.value(`dir +/ " ../samples/percussion/");
30  quit.open.allocs = allocMSGs.value(`dir +/ " ../samples/strings.open/");
31  quit.harm.allocs = allocMSGs.value(`dir +/ " ../samples/strings.harmonics/");
32  quit.mute.allocs = allocMSGs.value(`dir +/ " ../samples/strings.muted/");
33  quit.bass.allocs = allocMSGs.value(`dir +/ " ../samples/ostinato.bass/");
34
35  // execute everything in non real time, generate soundfile starting with 4 seconds of silence
36  Score.recordNRT(
37    perc.allocs[0] ++ quit.open.allocs[0] ++ quit.harm.allocs[0] ++ quit.mute.allocs[0] ++ quit.bass.allocs
38    [0] ++
39    [
40      [0, [\b.alloc, sample.index + 1, 10000, 5]],
41      [0, [\d.recv, `ostinato.and.interrupt.asBytes(s);]],
42      [4, [\s.new, \ostinato.and.interrupt, 10000, 0, 0, \scoreBuf, sample.index + 1, \seed, seed] ++
43        [\percBufDelims, $[,] ++ perc.allocs[1] ++ [$]] ++
44        [\quitOpenBufDelims, $[,] ++ quit.open.allocs[1] ++ [$]] ++
45        [\quitHarmBufDelims, $[,] ++ quit.harm.allocs[1] ++ [$]] ++
46        [\quitMuteBufDelims, $[,] ++ quit.mute.allocs[1] ++ [$]] ++
47        [\quitBassBufDelims, $[,] ++ quit.bass.allocs[1] ++ [$]]
48      ],
49    ,

```



```

45         [60 * 20, [\b_write, sample.index + 1, ~dir +/+ "gen_data_resources/ostinato_and_interrupt_data
46             .wav", "WAV", "float"]],
47         [60 * 20, [\c_set, 0, 0]]],
48     ~dir +/+ "gen_data_resources/ostinato_and_interrupt_osc",
49     ~dir +/+ "../audio/ostinato_and_interrupt.wav",
50     headerFormat: "WAV", options: ServerOptions.new.numOutputBusChannels = 26, action: {
51
52         // trim the multichannel audio file down to the correct size
53         var datasf, insf, outsf, data, durSum, n, pad, newSize;
54
55         ~appStatusString = "writing files";
56         datasf = SoundFile.openRead(~dir +/+ "gen_data_resources/ostinato_and_interrupt_data.wav");
57         datasf.readData(data = FloatArray.newClear(10000));
58         datasf.close;
59         data = data.clump(5).drop(1);
60         durSum = 0;
61         n = 0;
62         while( {data[n][4] != 0}, {
63             durSum = durSum + data[n][4];
64             n=n+1;
65         });
66         insf = SoundFile.openRead(~dir +/+ "../audio/ostinato_and_interrupt.wav");
67         outsf = SoundFile.new.headerFormat.(insf.headerFormat).numChannels.(
68             insf.numChannels).sampleRate.(insf.sampleRate).sampleFormat.(insf.sampleFormat);
69         outsf.openWrite(~dir +/+ "../audio/ostinato_and_interrupt_cut.wav");
70         pad = insf.sampleRate * 4 * insf.numChannels;
71         insf.readData(data = FloatArray.newClear(pad));
72         (((durSum + (22 * 5))/5.0).ceil).do({
73             insf.readData(data = FloatArray.newClear(insf.sampleRate * insf.numChannels));
74             outsf.writeData(data)});
75         insf.close;
76         outsf.close;
77         File.delete(~dir +/+ "../audio/ostinato_and_interrupt.wav");
78         File.copy(~dir +/+ "../audio/ostinato_and_interrupt_cut.wav", ~dir +/+ "../audio/
79             ostinato_and_interrupt.wav");
80         File.delete(~dir +/+ "../audio/ostinato_and_interrupt_cut.wav");
81
82         // call transcriber function
83         ~appStatusString = "generating lilypond";
84         ~generateLilypond.value;
85
86         // load the buffer for playback
87         ~appStatusString = "loading buffer";
88         Buffer.read(s, ~dir +/+ "../audio/ostinato_and_interrupt.wav", action: {
89             |buf| ~totalDur = buf.duration;
90             if(isLaunch == true, {~play = Synth.new(\play, [\buf, buf])}, {~play.set(\buf, buf)});
91             ~appStatusFunc.stop;
92             {~appStatus.string = "ready"}.defer
93         });
94     });
95
96 // uncomment below to run generator without gui
97 /*
98 "ostinato_and_interrupt_generator_synthdef.scd".loadRelative(true, {
99     ~dir = thisProcess.nowExecutingPath.dirname;
100     ~appStatus = StaticText();
101     ~generateData.value;
102 });
103 */

```

ostinato_and_interrupt_lilypond_generator_function.scd

```

1  (
2  //~FUNCTION THAT GENERATES THE LILYPOND FILES
3  ~generateLilypond = {
4      var sf, data, notes, perc, parts, noteNames, durSum, lastState, lastDur, n, beatPos, lastBassNote, lGrey, dGrey
5          , lilyRepeat, lilyTime;
6      var inFile, outFile, inString, outString;
7      sf = SoundFile.openRead(~dir +/+ "../supercollider/gen_data_resources/ostinato_and_interrupt_data.wav");
8      sf.readData(data = FloatArray.newClear(10000)); sf.close;
9      data = data.clump(5).drop(1);
10     notes = []; perc = []; parts = [[], [], [], [], [], [], []];
11     noteNames = ["d\'", "f\'", "a\'", "c\'\'", "e\'\'", "g\'\'"];
12     durSum = 60; lastState = -1; lastDur = -1; n = 0; beatPos = 0; lGrey = 70; dGrey = 50; lastBassNote = 11;
13     while( {data[n][4] != 0}, {
14         var state, string, note, sound, dur;
15         state = data[n][0]; string = data[n][1]; note = data[n][2]; sound = data[n][3]; dur = data[n][4];
16         data[n].postln; durSum = durSum + dur; durSum.postln;
17         if(lastState != state, {beatPos = 0}, {});
18         for(0, dur - 1, {
19             arg b;
20             var lilyStem, lilyRhythmMark, lilyStartBeam, lilyEndBeam, lilyBar, sec, minString, secString,
21                 lilyTime, lilyBracket, lilyNote;
22
23             lilyStem = case
24             {b == 0 && lastState == 0 && state == 1} {
25                 " \\override Staff.Stem #'transparent = ##f "
26             }
27             {b == 0 && ((lastState == 1 && state == 0) || (lastState == -1))} {
28                 " \\override Staff.Stem #'transparent = ##t "
29             }
30             {true} {" "};

```

```

28     lilyRhythmMark = case
29     {state == 0} {""}
30     {dur > 3} {""}
31     {lastState != state} {" " + dur.asString + " "}
32     {lastDur != dur} {" " + dur.asString + " "}
33     {true} {""};
34
35     lilyStartBeam = if(beatPos % 4 == 0, {" [ ", {""});
36     lilyEndBeam = if(beatPos % 4 == 3, {" ] ", {""});
37     beatPos = beatPos + 1;
38
39     lilyBar = if(b == 0 && (lastState != state), {' \\bar \\|\\' }, {""});
40
41     sec = ((durSum - dur) + b) / 5;
42     minString = (sec.trunc / 60).trunc.asString;
43     secString = (sec.trunc % 60).asString;
44     if(secString.size == 1, {secString = "0" + secString}, {});
45     lilyTime = case
46     {b == 0 && lastState == 0 && state == 1} || (sec % 12 == 0) {
47         ' \\mark \\markup{ \\fontsize #-2 \\' + minString + "' \\' + secString + '\\\\' } '
48     {b == 0 && ((lastState == 1 && state == 0) || (lastState == -1))} {""}
49     {true} {""};
50
51     lilyBracket = case
52     {b == 0 && state == 0 && string == 1 && (note != 7)} {
53         var res = if(lastBassNote != note, {"\\bracketify "}, {""}); lastBassNote = note; res;
54     {b == 0 && state == 0 && string == 0} {
55         var res = if(lastBassNote != note, {"\\bracketify "}, {""}); lastBassNote = note; res;
56     {true} {lastBassNote = lastBassNote; ""};
57
58
59     lilyNote = case
60     {(state == 0) && (b != 0)} {"s16 "}
61     {state == 0 && string == 0 && note == 0} {lilyBracket ++ noteNames[string] ++ "16 "}
62     {state == 0 && string == 1 && (note - 7) == 0} {lilyBracket ++ noteNames[string] ++ "16 "}
63     {(state == 0) && (string <= 1)} {
64         "\\once \\override NoteHead #\\stencil = #ly:text-interface:print " ++
65         "\\once \\override NoteHead #'text = \\markup { \\translate #'(-0 . -0.8) \\whiteout
66         \\pad-markup #0.5 " ++ '\\ ' ++
67         if(string == 0, {note.asString}, {(note - 7).asString}) ++ '\\ ' ++ lilyBracket ++
68         noteNames[string] ++ "16 "
69     {state == 0 && (string >= 2)} {noteNames[string] ++ "16 "}
70
71     {(state == 1) && (b != 0)} {"r16 " + lilyStartBeam + lilyEndBeam}
72     {state == 1 && sound == 0} {noteNames[string] ++ "16 " + lilyRhythmMark + lilyStartBeam +
73     lilyEndBeam}
74     {state == 1 && sound == 1} {noteNames[string] ++ "16 " + "\\harmonic " + lilyRhythmMark +
75     lilyStartBeam + lilyEndBeam}
76     {state == 1 && sound == 2} {"\\xNote " + noteNames[string] ++ "16 " + lilyRhythmMark +
77     lilyStartBeam + lilyEndBeam};
78
79     notes = notes.add(lilyBar ++ lilyTime ++ lilyStem ++ lilyNote);
80
81     perc = perc.add(if(sound == 2 || (state == 0),
82     {lilyBar ++ lilyTime ++ lilyStem ++ lilyNote},
83     {lilyBar ++ lilyTime ++ " \\once \\override Stem.color = #(x11-color 'grey" ++ lGrey.
84     asString ++ ") " ++ lilyStem ++
85     " \\once \\override NoteHead.color = #(x11-color 'grey" ++ lGrey.asString ++ "
86     ) " ++ lilyNote}));
87
88     for(0, 5, {
89         arg p; var grey;
90         grey = if(sound == 2, {dGrey}, {lGrey});
91         parts[p] = parts[p].add(if(string == p || (state == 0),
92         {lilyBar ++ lilyTime ++ lilyStem ++ lilyNote},
93         {lilyBar ++ lilyTime ++ " \\once \\override Stem.color = #(x11-color 'grey" ++
94         grey.asString ++ ") " ++ lilyStem ++
95         " \\once \\override NoteHead.color = #(x11-color 'grey" ++ grey.
96         asString ++ ") " ++ lilyNote}));
97     });
98     lastState = state;
99     lastDur = dur;
100     n = n + 1;
101 });
102
103 lilyRepeat = "\\repeat unfold " + ((durSum-60)/5/12).trunc.asString + " { \\repeat unfold 59 { s16 \\noBreak
104 } s16 \\break } ";
105 lilyTime = "\\time 60/16 ";
106
107 inFile = File("~/dir /+ " + "../lilypond/ostinato.and.interrupt.lilypond.score.template.ly", "r");
108 inString = inFile.readAllString;
109 inFile.close;
110
111 outFile = File("~/dir /+ " + "../lilypond/ostinato.and.interrupt.lilypond.guitar.part.ly", "w");
112 outString = "<< " + lilyRepeat + lilyTime +
113     "{ \\override Staff.Rest #'transparent = ##t " + notes.join ++ ' \\bar \\|\\.\\' } >>';
114 outFile.write(inString.replace("<<music>>", outString).replace("piece = " ++ '\\part\\', "piece = " ++ '\\
115 guitar/all\\'));
116 outFile.close;
117
118 outFile = File("~/dir /+ " + "../lilypond/ostinato.and.interrupt.lilypond.percussion.part.ly", "w");
119 outString = "<< " + lilyRepeat + lilyTime +
120     "{ \\override Staff.Rest #'transparent = ##t " + perc.join ++ ' \\bar \\|\\.\\' } >>';

```

```

111     outFile.write(inString.replace("%<<music>>", outString).replace("piece = " ++ "\"part\"", "piece = " ++ "\"
112         percussion\""));
113     outFile.close;
114
115     for(0, 5, {
116         arg p, staff;
117         outFile = File("~/lilypond/ostinato_and_interrupt_lilypond_ensemble_part_" ++ (6 - p).
118             asString ++ ".ly", "w");
119         staff = Array.fill(6, {|i| if(i == (5 - p), {"#f"}, {"(x11-color 'grey" ++ lGrey.asString ++ ") "})});
120         staff = "\\override Staff.StaffSymbol.stencil = #(color-staff-lines " ++ staff ++ ")";
121         outString = "<< " ++ lilyRepeat ++ lilyTime ++
122             "{ " ++ staff ++ " \\override Staff.Rest #'transparent = ##t " ++ (parts[p]).join ++ ' "\\bar \"|.\"
123             } >>';
124         outFile.write(inString.replace("%<<music>>", outString).replace(
125             "piece = " ++ "\"part\"", "piece = " ++ "\"ensemble part " ++ (6 - p).asString ++ "\""));
126         outFile.close;
127     });
128
129     // uncomment below generate lilypond files without gui (requires resources to exist)
130     /*(
131     ^dir = thisProcess.nowExecutingPath.dirname;
132     ^generateLilypond.value
133     )*/

```

ostinato_and_interrupt_player_synthdef.scd

```

1  (
2  //~~SYNTHDEF THAT PLAYS THE PIECE AND ACCEPTS CONTROL FROM THE GUI
3  SynthDef(\play, {
4      arg buf = 0, env, playRate = 0, startPos = 0, startTrig = 0, curDur,
5      goVol = #[1, 1, 1, 1, 1, 1], giVol = #[1, 1, 1, 1, 1, 1], pVol = #[1, 1, 1, 1, 1, 1], eVol = #[1, 1, 1, 1, 1, 1],
6      aVol = #[1, 1, 1, 1, 1, 1],
7      goMute = #[1, 1, 1, 1, 1, 1], giMute = #[1, 1, 1, 1, 1, 1], eMute = #[1, 1, 1, 1, 1, 1], pMute = #[1, 1, 1, 1, 1, 1],
8      aMute = #[1, 1, 1, 1, 1, 1],
9      goPan = #[0, 0, 0, 0, 0, 0], giPan = #[0, 0, 0, 0, 0, 0], ePan = #[0, 0, 0, 0, 0, 0], pPan = #[0, 0, 0, 0, 0, 0],
10     aPan = #[0, 0, 0, 0, 0, 0],
11     masterVolGroups = #[1, 1, 1, 1, 1], masterMuteGroups = #[1, 1, 1, 1, 1],
12     allMasterVol = 1, allMasterMute = 1;
13     var phasor, player;
14     var guitarOTracks, guitarITracks, percussionTracks, ensembleTracks, accompTracks;
15     var guitarOTracksPanned, guitarITracksPanned, percussionTracksPanned, ensembleTracksPanned, accompTracksPanned;
16     var guitarOMaster, guitarIMaster, percussionMaster, ensembleMaster, accompMaster;
17     var allMaster;
18     var imp, delimp;
19
20     player = PlayBuf.ar(26, buf, playRate, startTrig, startPos * BufFrames.kr(buf));
21     phasor = Phasor.ar(startTrig,
22         Select.kr(playRate, [0, BufRateScale.kr(buf)]),
23         0, BufFrames.kr(buf), startPos * BufFrames.kr(buf));
24
25     guitarOTracks = { |i| var string = 5 - i; player[string - 2] * goVol[i] * goMute[i] } ! 4;
26     guitarITracks = { |i| var string = 5 - i; player[string + 4] * giVol[i] * giMute[i] } ! 6;
27     percussionTracks = { |i| var string = 5 - i; player[string + 10] * pVol[i] * pMute[i] } ! 6;
28     ensembleTracks = { |i| var string = 5 - i; player[string + 16] * eVol[i] * eMute[i] } ! 6;
29     accompTracks = { |i| player[i + 22] * aVol[i] * aMute[i] } ! 4;
30
31     guitarOTracksPanned = { |i| Pan2.ar(guitarOTracks[i], goPan[i]) } ! 4;
32     guitarITracksPanned = { |i| Pan2.ar(guitarITracks[i], giPan[i]) } ! 6;
33     percussionTracksPanned = { |i| Pan2.ar(percussionTracks[i], pPan[i]) } ! 6;
34     ensembleTracksPanned = { |i| Pan2.ar(ensembleTracks[i], ePan[i]) } ! 6;
35     accompTracksPanned = { |i| Pan2.ar(accompTracks[i], aPan[i]) } ! 4;
36
37     guitarOMaster = Mix.new(guitarOTracksPanned) * masterVolGroups[0] * masterMuteGroups[0];
38     guitarIMaster = Mix.new(guitarITracksPanned) * masterVolGroups[1] * masterMuteGroups[1];
39     percussionMaster = Mix.new(percussionTracksPanned) * masterVolGroups[2] * masterMuteGroups[2];
40     ensembleMaster = Mix.new(ensembleTracksPanned) * masterVolGroups[3] * masterMuteGroups[3];
41     accompMaster = Mix.new(accompTracksPanned) * masterVolGroups[4] * masterMuteGroups[4];
42
43     allMaster = Mix.new([guitarOMaster, guitarIMaster, percussionMaster, ensembleMaster, accompMaster]) *
44         allMasterVol * allMasterMute;
45     Out.ar(0, allMaster);
46
47     curDur = ((A2K.kr(phasor) / BufFrames.kr(buf)) * BufDur.kr(buf) * 5).trunc;
48     //Optional click - uncomment and send to an output not used to give the guitarist a click track.
49     //Out.ar(2, 10 * BPF.ar(WhiteNoise.ar * EnvGen.kr(Env.perc(0.01, 0.1), curDur % 4 <= 0), 440 * ((curDur % 20 <=
50         0) + 1), 0.02));
51     SendTrig.kr(Changed.kr(curDur), 0, curDur);
52     imp = Impulse.kr(10);
53     delimp = Delay1.kr(imp);
54     SendReply.kr(imp,
55         '/allMasterLevels',
56         values: [Amplitude.kr(allMaster)]);
57     SendReply.kr(imp,
58         '/groupMasterLevels',
59         values: [
60             Amplitude.kr(guitarOMaster) ++ Amplitude.kr(guitarIMaster) ++
61             Amplitude.kr(percussionMaster) ++ Amplitude.kr(ensembleMaster) ++ Amplitude.kr(accompMaster)];
62     SendReply.kr(imp,
63         '/groupTrackLevels',
64         values: [Amplitude.kr(guitarOTracks) ++ Amplitude.kr(guitarITracks) ++

```

```

60         Amplitude.kr(percussionTracks) ++ Amplitude.kr(ensembleTracks) ++ Amplitude.kr(accompTracks));
61     }).add;
62 }

```

ostinato_and_interrupt_gui_generator_function.scd

```

1  (
2  //~FUNCTION THAT GENERATES THE GUI
3  ~generateGUI = {
4      var win, clockStringFunc, metronomeStringFunc, metronomeColorFunc, masterView, faderViews, tabs;
5      var tabButtonReset, masterButton, guitarOButton, guitarIButton, percButton, ensembleButton, accompButton,
6          startPos = 0;
7      var groupNames = ["guitar - ostinato", "guitar - interrupt", "percussion", "interrupt highlights", "fields /
8          beats / flicker"], groupAbbr = ["go", "gi", "p", "e", "a"];
9      var accompNames = ["brown noise", "white noise", "sine beating", "flicker"];
10     var goVol, giVol, pVol, eVol, aVol, goPan, giPan, pPan, ePan, aPan, goMute, giMute, pMute, eMute, aMute,
11         volGroups, panGroups, muteGroups;
12     var masterMuteGroups, masterVolGroups;
13
14     goVol = giVol = pVol = eVol = aVol = [0.8, 0.8, 0.8, 0.8, 0.8, 0.8];
15     goMute = giMute = pMute = eMute = aMute = [1, 1, 1, 1, 1, 1];
16     goPan = giPan = pPan = ePan = aPan = [0, 0, 0, 0, 0, 0];
17     volGroups = [goVol, giVol, pVol, eVol, aVol];
18     muteGroups = [goMute, giMute, pMute, eMute, aMute];
19     panGroups = [goPan, giPan, pPan, ePan, aPan];
20     masterMuteGroups = [1, 1, 1, 1, 1];
21     masterVolGroups = [0.8, 0.8, 0.8, 0.8, 0.8];
22
23     clockStringFunc = {
24         arg div;
25         var min, sec;
26         sec = (div / 5).trunc;
27         min = (sec / 60).trunc.asString;
28         if(min.size == 1, {min = "0" ++ min}, {});
29         sec = (sec % 60).trunc.asString;
30         if(sec.size == 1, {sec = "0" ++ sec}, {});
31         min ++ ":" ++ sec
32     };
33     metronomeStringFunc = { arg div; case {div % 20 < 2} {" " } {div % 4 < 2} {" " } {true} {" " } };
34     metronomeColorFunc = { arg div; case {div % 20 < 2} {Color.red} {div % 4 < 2} {Color.blue} {true} {Color.black}
35         };
36
37     ~appStatusFunc = Task({
38         loop {
39             {~appStatus.string = ~appStatusString ++ "*"}.defer;
40             0.5.wait; {~appStatus.string = ~appStatusString ++ "* *"}.defer;
41             0.5.wait; {~appStatus.string = ~appStatusString ++ "* * *"}.defer;
42             0.5.wait; {~appStatus.string = ~appStatusString ++ "* * * *"}.defer;
43             0.5.wait; {~appStatus.string = ~appStatusString ++ "* * * * *"}.defer;
44             0.5.wait;
45         }
46     });
47
48     win = Window("ostinato and interrupt", Rect(500, 500, 1100, 500), false).front;
49     masterView = {
50         var view, masterIndicators, master, generator, transport, ranSeed, startPosText, pauseButton, clock,
51             metronome;
52
53         OSCFunc({ arg msg, time; {
54             clock.string = clockStringFunc.value(msg[3]);
55             metronome.stringColor = metronomeColorFunc.value(msg[3]);
56             metronome.string = metronomeStringFunc.value(msg[3])}.defer;
57         }, '/tr', s.addr);
58         OSCFunc({arg msg; {
59             {[i] masterIndicators[i].value = msg[3 + i].ampdb.linlin(-40, 0, 0, 1)} ! 2}.defer,
60             '/allMasterLevels', s.addr);
61
62         view = View(win);
63         masterIndicators = [LevelIndicator(), LevelIndicator()];
64         master = HLayout(
65             VLayout(
66                 HLayout(
67                     Slider(view).value.(0.8).action-({|v| ~play.set(\allMasterVol, v.value * 1.25)
68                         },
69                     masterIndicators[0], masterIndicators[1]),
70                     Button(view).states-([["mute", Color.black], ["mute", Color.black, Color.grey]]).
71                         action-({
72                             |[v| ~play.set(\allMasterMute, (1 - v.value).abs)},
73                             StaticText(view).string-("master")
74                         ).align-(\center),
75                     StaticText(view).string-("all").align-(\center)),
76                     nil);
77         generator = HLayout(
78             Button(view).states-([["generate"]]).action-({
79                 ~appStatusString = "generating data";
80                 ~appStatusFunc.start;
81                 ~generatedData.value(seed: ranSeed.string.asInteger);
82             },
83             ranSeed = TextField(view).string-("20170121"),
84             Button(view).states-([["reset seed"]]).action-({ ranSeed.string = "20170121"}),
85             Button(view).states-([["random seed"]]).action-({ ranSeed.string = 50000000.rand.asString}),
86             [~appStatus = StaticText(view).string-("status: ready"), stretch: 1],
87             nil);
88         transport = HLayout(

```

```

81         Button(view).states-([[ "play", Color.black], [ "stop", Color.black, Color.grey]]).action-(
82             [| pState |
83                 pauseButton.value = 0;
84                 if(pState.value == 0, {~play.set(\playRate, 0, \startTrig, 0);
85                     clock.string = clockStringFunc.value((startPos * ~totalDur * 5).trunc)
86                     },
87                     {~play.set(\startPos, startPos, \playRate, 1, \startTrig, 1)})),
88             pauseButton = Button(view).states-([[ "pause", Color.black], [ "pause", Color.black, Color.grey
89             ]]).action-(
90                 [| pState |
91                     if(pState.value == 1, {~play.set(\playRate, 0)}, {~play.set(\playRate, 1)})),
92                 StaticText(view).string-"start time",
93                 [Slider(view, Rect(0, 0, 30, 5)).action-(
94                     [|pos|
95                         var min, sec;
96                         startPosText.string = clockStringFunc.value((pos.value * ~totalDur * 5).trunc);
97                         startPos = pos.value;
98                     ]), stretch: 1],
99                 startPosText = StaticText(win).string-"00:00").font-(Font("Monaco", 15)),
100                 nil);
101         view.layout-(HLayout(master,
102             [VLayout(generator, nil,
103                 HLayout(clock = StaticText(win).string-"00:00").font-(Font("Monaco", 200)),
104                 StaticText(win).string-"|").font-(Font("Monaco", 200)),
105                 metronome = StaticText(win).string-" ").font-(Font("Monaco", 300)).stringColor-(
106                     Color.red)),
107                 nil, transport
108             ), alignment: \top])));
109         faderViews = { |group|
110             var view, masterIndicators, trackIndicators, master, tracks;
111             view = View(win);
112             masterIndicators = {LevelIndicator()} ! 10;
113             trackIndicators = {LevelIndicator()} ! 26;
114             OSCFunc.new({arg msg; {
115                 [|i| masterIndicators[i].value = msg[3 + i].ampdb.linlin(-40, 0, 0, 1)} ! 10}.defer},
116                 '/groupMasterLevels', s.addr);
117             OSCFunc.new({arg msg; {
118                 [|i| trackIndicators[i].value = msg[3 + i].ampdb.linlin(-40, 0, 0, 1)} ! 26}.defer},
119                 '/groupTrackLevels', s.addr);
120             master = HLayout(
121                 VLayout(
122                     [HLayout(
123                         Slider(view).value_(0.8).action-(
124                             [|v| masterVolGroups[group] = v.value * 1.25; ~play.set(
125                                 \masterVolGroups, masterVolGroups)}),
126                             masterIndicators[group * 2],
127                             masterIndicators[group * 2 + 1]), stretch: 2],
128                         Button(view).states-([[ "mute", Color.black], [ "mute", Color.black, Color.grey]]).
129                         action-(
130                             [|v| masterMuteGroups[group] = (1 - v.value).abs; ~play.set(\masterMuteGroups,
131                                 masterMuteGroups)}),
132                         StaticText(view).string-"master").align-(\center),
133                         StaticText(view).string-"( ++groupNames[group]++)").align-(\center)
134                     ),
135                     nil);
136             tracks = { |part|
137                 HLayout(
138                     VLayout(
139                         HLayout(
140                             Slider(view).value_(0.8).action-(
141                                 [|v| volGroups[group][part] = v.value * 1.25; ~play.set(
142                                     groupAbbr[group] ++ "Vol", volGroups[group])}),
143                                 trackIndicators[group * 6 + part - if(group > 0, {2}, {0})]),
144                                 Button(view).states-([[ "mute", Color.black], [ "mute", Color.black, Color.grey
145                                 ]]).action-(
146                                     [|v| muteGroups[group][part] = (1 - v.value).abs; ~play.set(groupAbbr[
147                                         group] ++ "Mute", muteGroups[group])}),
148                                     StaticText(view).string-"pan").align-(\center),
149                                     Knob(view).value_(0.5).action-(
150                                         [|v| panGroups[group][part] = v.value * 2 - 1; ~play.set(groupAbbr[
151                                             group] ++ "Pan", panGroups[group])}),
152                                     StaticText(view).string-(
153                                         if(group == 4, {accompNames[part]}, {(6 - part).asString})).align-(
154                                             \center)
155                                 ),
156                                 nil)
157                             } ! if((group == 0) || (group == 4), {4}, {6});
158                             view.layout-(HLayout(master, nil, *tracks)) ! 5;
159                             tabButtonReset = {masterButton.value = 1;
160                                 guitarOButton.value = 1; guitarIBButton.value = 1; percButton.value = 1; ensembleButton.value = 1;
161                                 accompButton.value = 1};
162                             win.layout = VLayout(
163                                 HLayout(
164                                     masterButton = Button().states-([[ "master controls", Color.white, Color.grey], [ "master
165                                     controls", Color.black]]).action-(
166                                         {tabButtonReset.value; masterButton.value = 0; tabs.index = 0 }).value_(0),
167                                     guitarOButton = Button().states-([[ "guitar (ostinato)", Color.white, Color.grey], [ "guitar (
168                                     ostinato)", Color.black]]).action-(
169                                         {tabButtonReset.value; guitarOButton.value = 0; tabs.index = 1 }).value_(1),
170                                     guitarIBButton = Button().states-([[ "guitar (interrupt)", Color.white, Color.grey], [ "guitar (
171                                     interrupt)", Color.black]]).action-(
172                                         {tabButtonReset.value; guitarIBButton.value = 0; tabs.index = 2 }).value_(1),

```

```

160         percButton = Button().states-([["percussion", Color.white, Color.grey], ["percussion", Color.
161             black])).action.(
162             {tabButtonReset.value; percButton.value = 0; tabs.index = 3 }).value_(1),
163         ensembleButton = Button().states-([["interrupt highlights", Color.white, Color.grey], ["
164             interrupt highlights", Color.black])).action.(
165             {tabButtonReset.value; ensembleButton.value = 0; tabs.index = 4 }).value_(1),
166         accompButton = Button().states-([["fields / beats / flicker", Color.white, Color.grey],
167             ["fields / beats / flicker", Color.black])).action.(
168             {tabButtonReset.value; accompButton.value = 0; tabs.index = 5 }).value_(1)),
169         tabs = StackLayout(masterView.value, faderViews[0], faderViews[1], faderViews[2], faderViews[3],
            faderViews[4]));
    };
}

```

ostinato_and_interrupt_lilypond_score_template.ly

```

1 \version "2.18.2"
2 \paper {
3   top-system-spacing =
4   #'((basic-distance . 15 )
5     (minimum-distance . 15 )
6     (padding . 0 )
7     (stretchability . 0))
8
9   #(set-paper-size "a4" 'portrait)
10  system-system-spacing =
11  #'((basic-distance . 24)
12    (minimum-distance . 24)
13    (padding . 0)
14    (stretchability . 0))
15
16  min-systems-per-page = 8
17  max-systems-per-page = 8
18
19  print-page-number = ##t
20  oddHeaderMarkup = \markup \fill-line { " " }
21  evenHeaderMarkup = \markup \fill-line { " " }
22  oddFooterMarkup = \markup {
23    \fill-line {
24      \on-the-fly #not-first-page
25      \concat {
26        "_"
27        \fontsize #1.5
28        \on-the-fly #print-page-number-check-first
29        \fromproperty #'page:page-number-string
30        "_"
31      }
32    }
33  }
34  evenFooterMarkup = \markup {
35    \on-the-fly #not-first-page
36    \fill-line {
37      \concat {
38        "_"
39        \fontsize #1.5
40        \on-the-fly #print-page-number-check-first
41        \fromproperty #'page:page-number-string
42        "_"
43      }
44    }
45  }
46 }
47 \header {
48   title = \markup { \normal-text \italic {ostinato and interrupt}}
49   piece = "part"
50   opus = \markup { \concat {"version generated: " } #(strftime "%Y.%m.%d" (localtime (current-time)))}
51   composer = "michael winter (mexico city, mx; 2017)"
52   tagline = ""
53 }
54 #(set-global-staff-size 16)
55 \layout {
56   indent = 0.0\cm
57   ragged-right = ##t
58   \context {
59     \Staff
60     \override StaffSymbol.line-count = #6
61     \override StaffSymbol.staff-space = #1.8
62     \override Beam.positions = #'(-5 . -5)
63     \override Stem.direction = #DOWN
64     \override Stem.stemlet-length = #1
65     \override Beam.breakable = ##t
66     \remove "Clef-engraver"
67     \remove "Time-signature-engraver"
68   }
69 }
70
71 #(define-public ((color-staff-lines . rest) grob)
72
73   (define (index-cell cell dir)
74     (if (equal? dir RIGHT)
75         (cdr cell)
76         (car cell)))
77

```

```

78 (define (index-set-cell! x dir val)
79   (case dir
80     ((-1) (set-car! x val))
81     ((1) (set-cdr! x val))))
82
83 (let* ((common (ly:grob-system grob))
84        (span-points '(0 . 0))
85        (thickness (* (ly:grob-property grob 'thickness 1.0)
86                      (ly:output-def-lookup (ly:grob-layout grob) 'line-thickness)))
87        (width (ly:grob-property grob 'width))
88        (line-positions (ly:grob-property grob 'line-positions))
89        (staff-space (ly:grob-property grob 'staff-space 1))
90        (line-stencil #f)
91        (total-lines empty-stencil)
92        ;; use a local copy of colors list, since
93        ;; stencil creation mutates list
94        (colors rest))
95
96   (for-each
97     (lambda (dir)
98       (if (and (= dir RIGHT)
99               (number? width))
100         (set-cdr! span-points width)
101         (let* ((bound (ly:spanner-bound grob dir))
102                (bound-ext (ly:grob-extent bound bound X)))
103
104           (index-set-cell! span-points dir
105                             (ly:grob-relative-coordinate bound common X))
106           (if (and (not (ly:item-break-dir bound))
107                   (not (interval-empty? bound-ext)))
108               (index-set-cell! span-points dir
109                                 (+ (index-cell span-points dir)
110                                   (index-cell bound-ext dir))))))
111         (index-set-cell! span-points dir (- (index-cell span-points dir)
112                                              (* dir thickness 0.5))))
113     (list LEFT RIGHT))
114
115   (set! span-points
116         (coord-translate span-points
117                           (- (ly:grob-relative-coordinate grob common X))))
118   (set! line-stencil
119         (make-line-stencil thickness (car span-points) 0 (cdr span-points) 0))
120
121   (if (pair? line-positions)
122       (for-each (lambda (position)
123                   (let ((color (if (pair? colors)
124                                     (car colors)
125                                     #f)))
126                     (set! total-lines
127                           (ly:stencil-add
128                             total-lines
129                             (ly:stencil-translate-axis
130                               (if (color? color)
131                                   (ly:stencil-in-color line-stencil
132                                                         (first color)
133                                                         (second color)
134                                                         (third color))
135                               (* position staff-space 0.5) Y)))
136                     (and (pair? colors)
137                           (set! colors (cdr colors)))))
138                   line-positions)
139       (let* ((line-count (ly:grob-property grob 'line-count 5))
140              (height (* (1- line-count) (/ staff-space 2)))
141              (do ((i 0 (1+ i)))
142                  ((= i line-count))
143                  (let ((color (if (and (pair? colors)
144                                         (> (length colors) i))
145                                  (list-ref colors i)
146                                  #f)))
147                    (set! total-lines (ly:stencil-add
148                                        total-lines
149                                        (ly:stencil-translate-axis
150                                          (if (color? color)
151                                              (ly:stencil-in-color line-stencil
152                                                                    (first color)
153                                                                    (second color)
154                                                                    (third color))
155                                          line-stencil)
156                                        (- height (* i staff-space) Y))))))
157         total-lines))
158
159   #)
160 #)
161 (define-public (bracket-stencils grob)
162   (let ((lp (grob-interpret-markup grob (markup #:fontsize 3.5 #:translate (cons -0.3 -0.5) "[")
163                                     (rp (grob-interpret-markup grob (markup #:fontsize 3.5 #:translate (cons -0.3 -0.5) "]")
164                                     (list lp rp))))))
165     (list lp rp)))
166
167 bracketify = #)
168 (define-music-function (parser loc arg) (ly:music?)
169   (.i "Tag @var{arg} to be parenthesized.")
170   #)
171

```

```
172 {
173   \new Score
174   \with {
175     \remove "Bar-number-engraver"
176     proportionalNotationDuration = #(ly:make-moment 1 16)
177   }
178   <<
179   \new Staff
180   \with {
181     \remove "Stem-engraver"
182   }
183   %<<music>>
184   >>
185 }
```