

## *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 <sup>#</sup> −14¢	f <sup>#</sup> −14¢	f <sup>#</sup> −14¢	d	d	d
d	d	d	a	a	a
c	b	a <sup>#</sup>	f	e	d

\*Note that the f<sup>#</sup> is flat 14 cents, or hundredths of a tempered semitone, from the tempered f<sup>#</sup> 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<sup>#</sup> −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 mimicking the synthesis process by playing 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. Ideally, if 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 <sup>♯</sup> -14¢	c -31¢	g <sup>♯</sup> -49¢	a <sup>♯</sup> +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 and a timer 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).

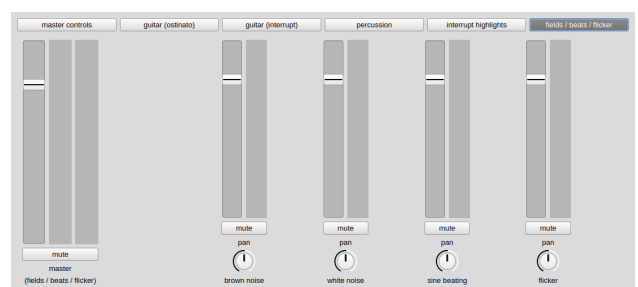
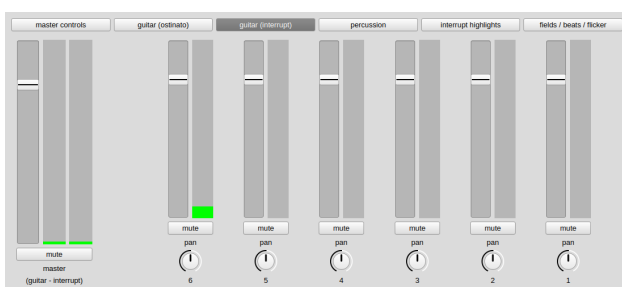
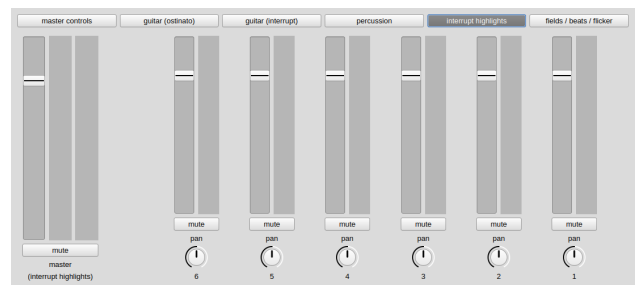
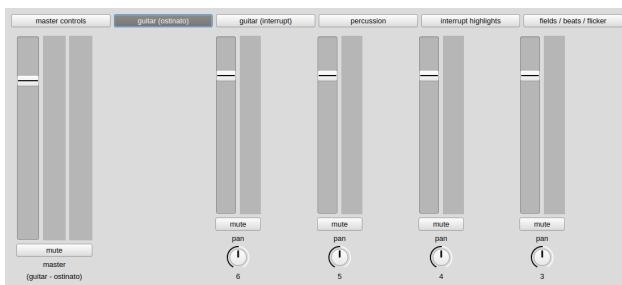
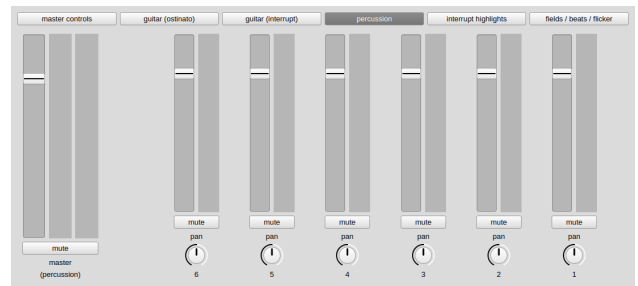
**strings\_open/**: each folder contains samples of the given open string from highest (1) to lowest (6).

**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](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](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: 2017.09.04

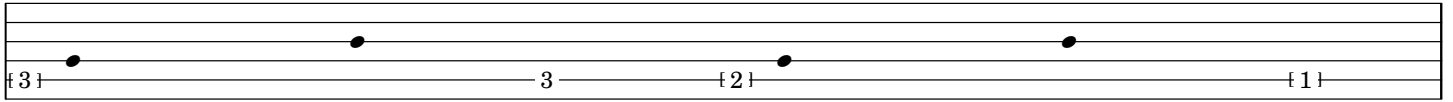
# *ostinato and interrupt*

guitar/all

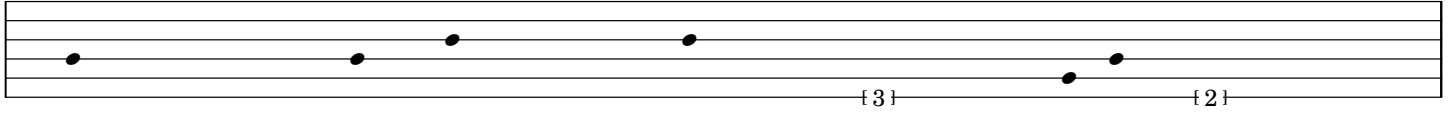
michael winter (mexico city, mx; 2017)

version generated: 2017.08.23

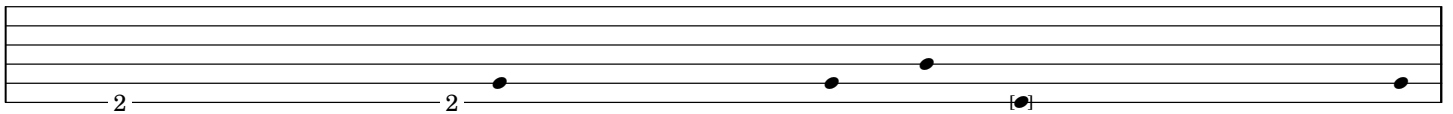
0'12"



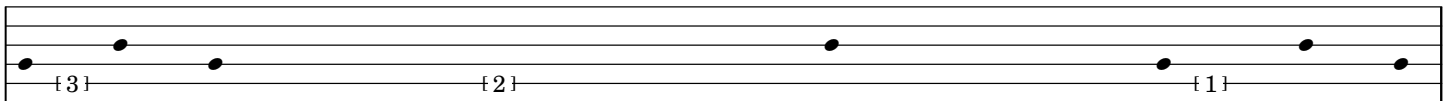
0'24"



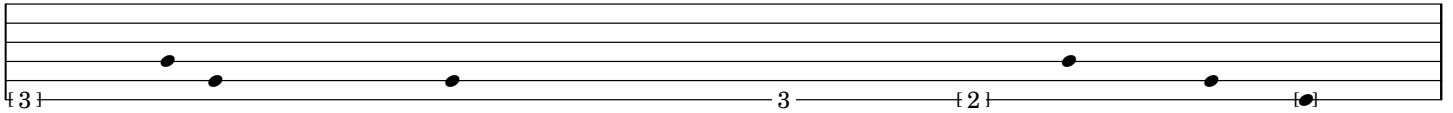
0'36"



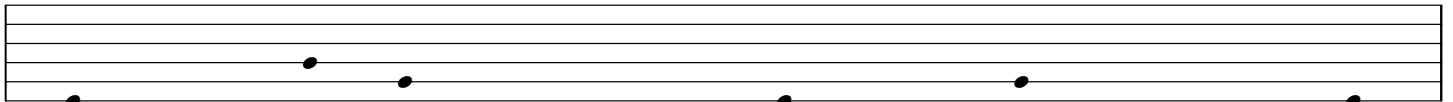
0'48"



1'00"

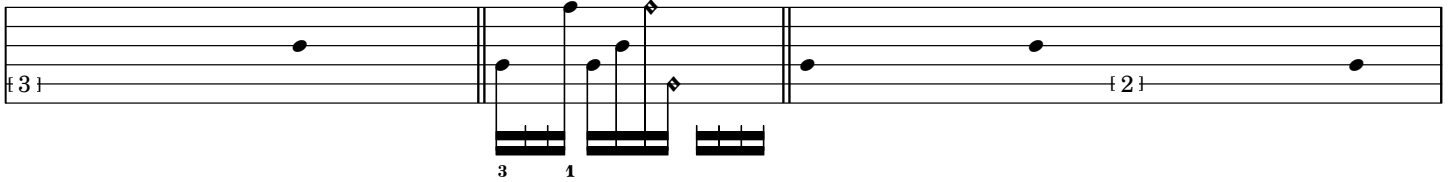


1'12"

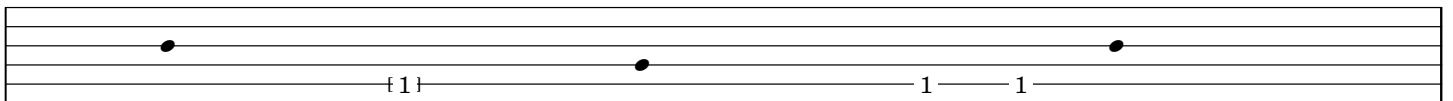


1'24"

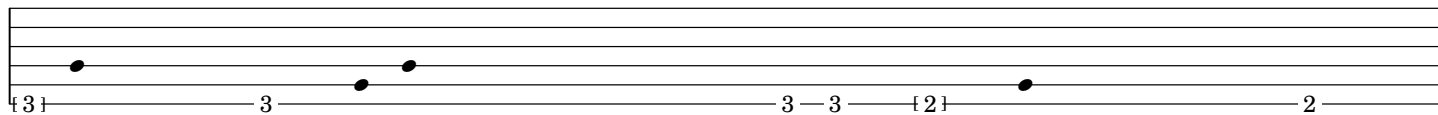
1'28"



1'36"

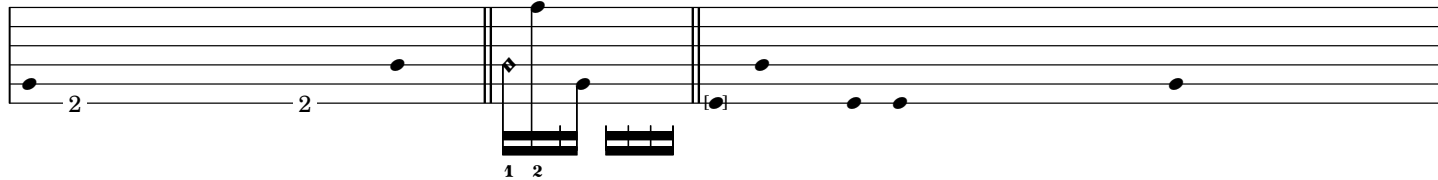


1'48"

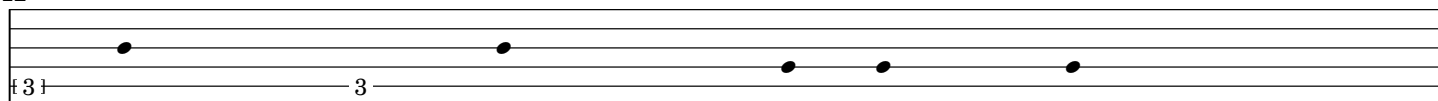


2'00"

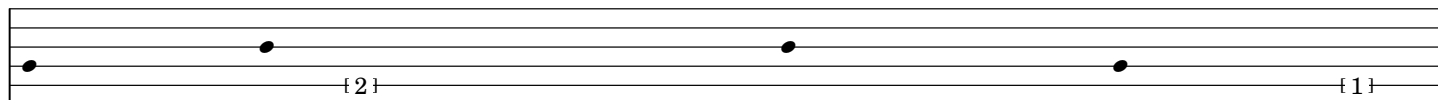
2'04"



2'12"

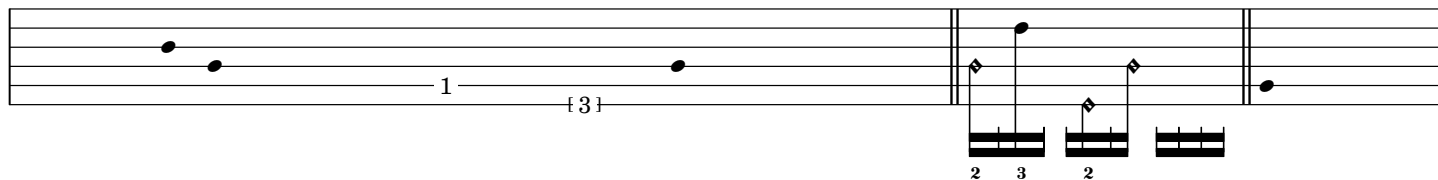


2'24"

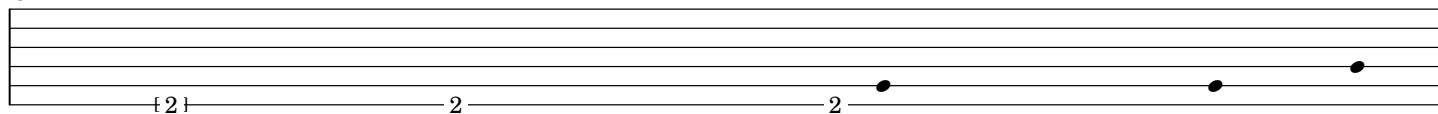


2'36"

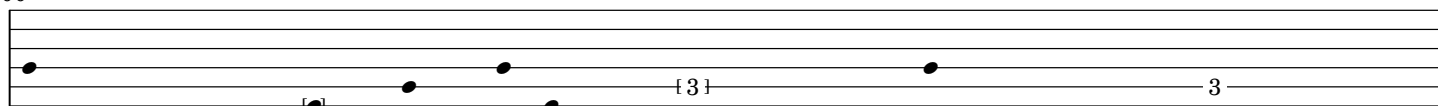
2'44"



2'48"

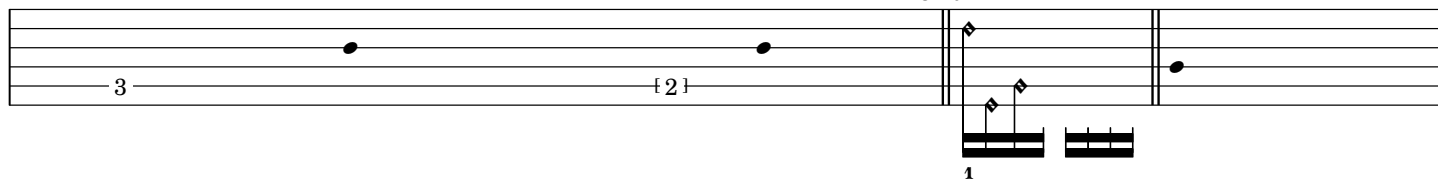


3'00"

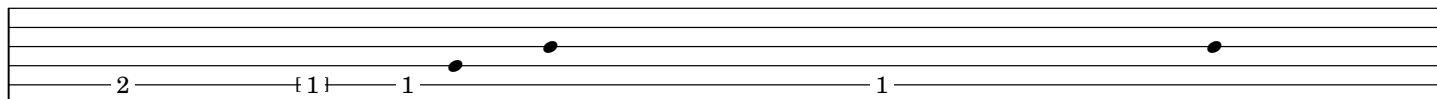


3'12"

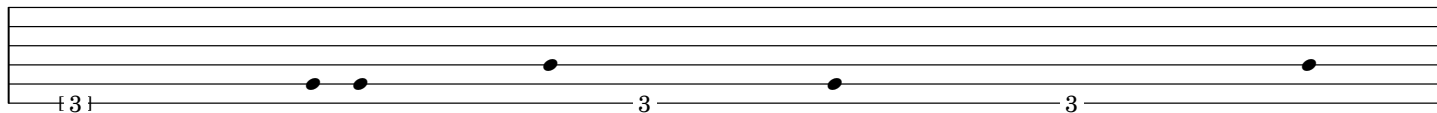
3'20"



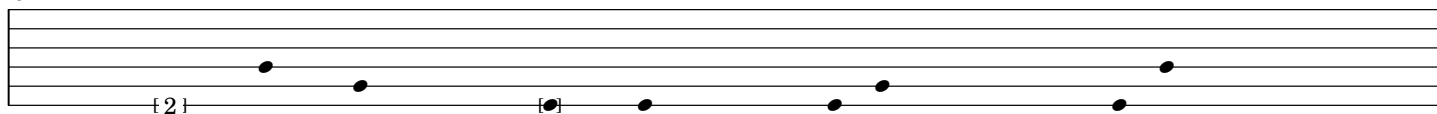
3'24"



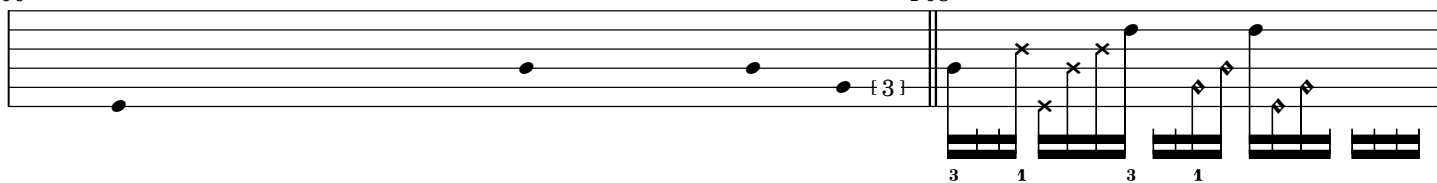
3'36"



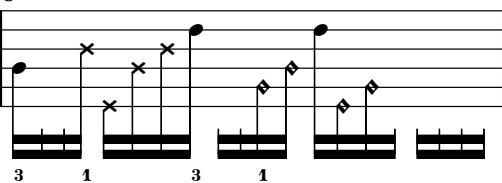
3'48"



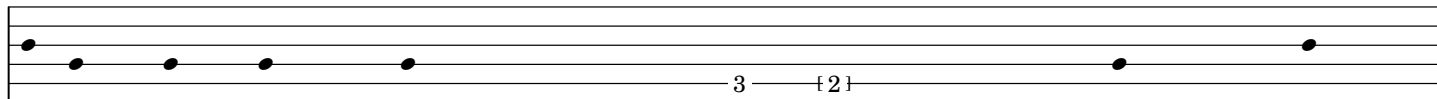
4'00"



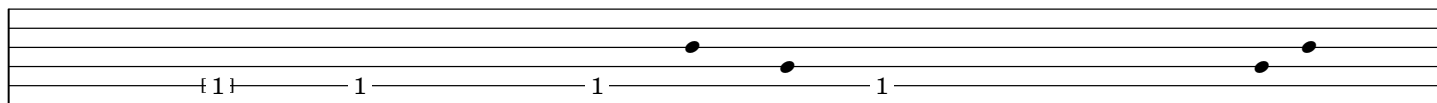
4'08"



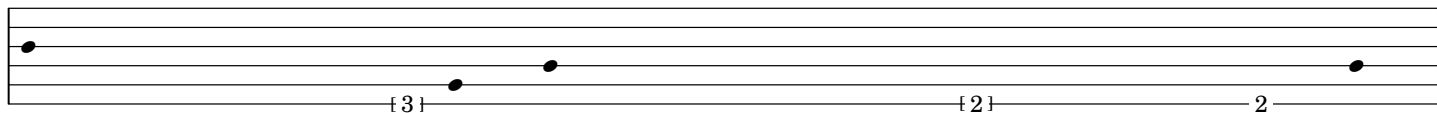
4'12"



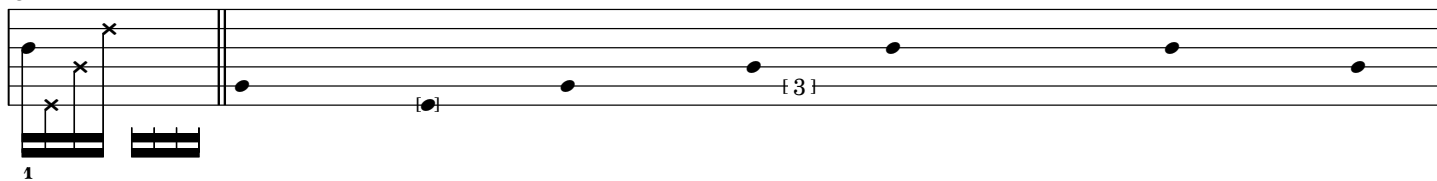
4'24"



4'36"



4'48"



[illegible]

5'24"

1 1 {3} 3 3

[illegible]

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 '3/4'. The score is divided into two systems. The first system contains the first line of the melody, which ends with a double bar line. The second system contains the second line of the melody, which also ends with a double bar line. The melody is written in a simple, folk-like style, using eighth and quarter notes. The lyrics 'The Rose Tree' are written below the staff, aligned with the notes. The score is marked with '6'12"' and '6'16"' above the staff, indicating the time taken to play each line. The score is marked with '1' and '2' below the staff, indicating the first and second endings. The score is marked with '1' and '2' below the staff, indicating the first and second endings. The score is marked with '1' and '2' below the staff, indicating the first and second endings.



A musical staff with three black dots on the first line and the label  $\{3\}$  at the end.

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 sequence of notes: a whole note G4, a whole note G4, a half note F#4, a half note E4, a quarter note D4, and a quarter note C4. The lower staff is a bass clef with a key signature of one flat. It contains a triplet of eighth notes (G3, A3, B3) marked with a '3' and a brace, followed by a quarter note G3, and then a triplet of eighth notes (F#3, E3, D3) marked with a '3' and a brace. The system ends with a double bar line.

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 fingerings (2, 2) and a triplet of eighth notes.

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 triplet of eighth notes.

8'48"

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

9'00"

9'04"

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

9'12"

Musical notation for 9'12". The staff shows a sequence of notes with fingerings (2, 2) and a triplet of eighth notes.

9'24"

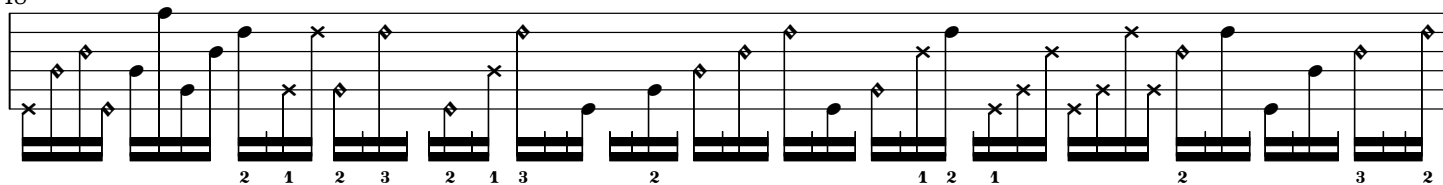
Musical notation for 9'24". The staff shows a sequence of notes with fingerings (2, 2) and a triplet of eighth notes.

9'36"

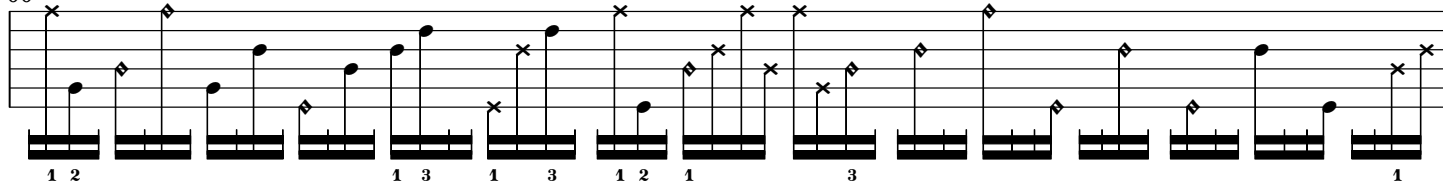
9'44"

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

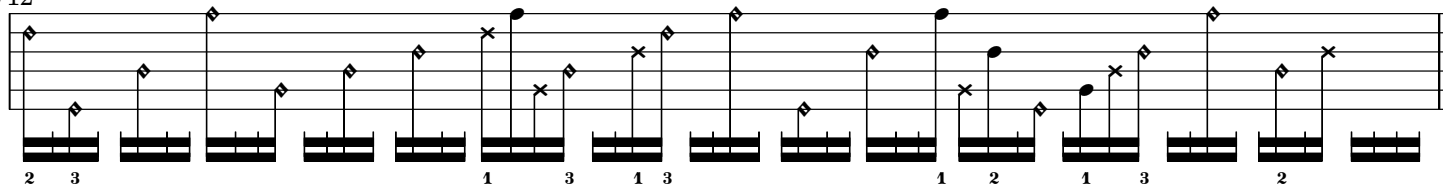
9'48"



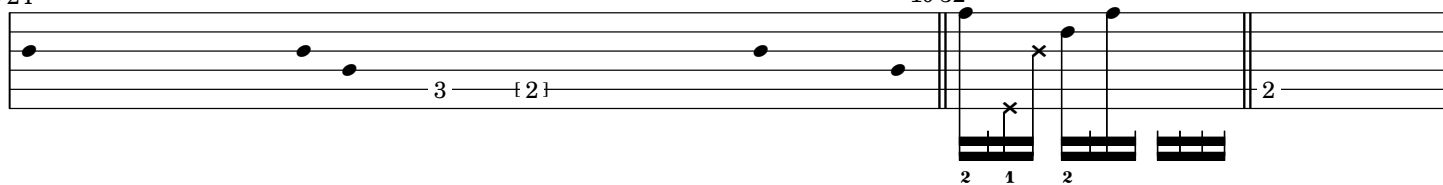
10'00"



10'12"

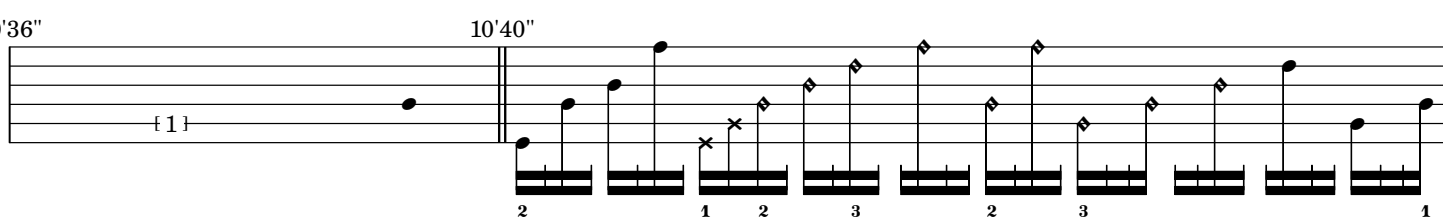


10'24"



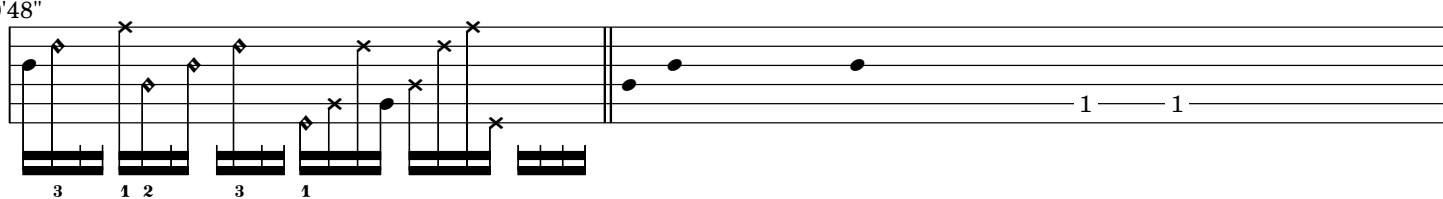
10'32"

10'36"

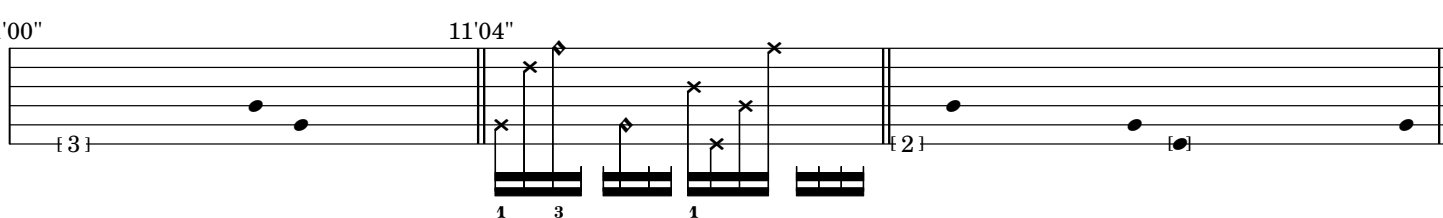


10'40"

10'48"

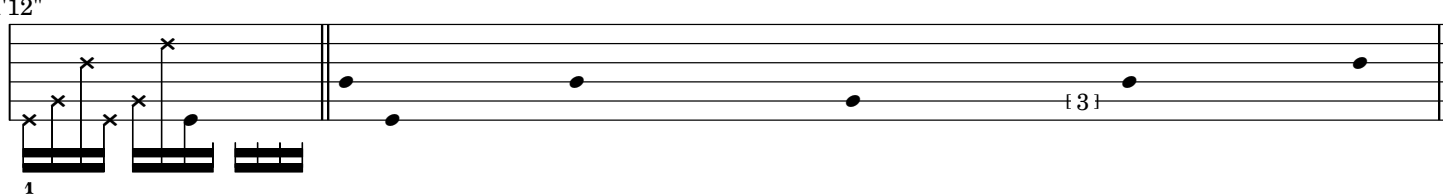


11'00"



11'04"

11'12"



11'24"


11'36"

11'48"

12'00"

12'12"

12'24"

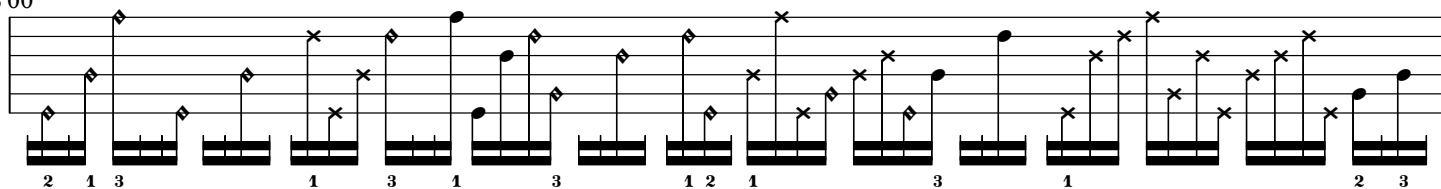


The musical score for '24" is written on a single staff. It begins with a treble clef and a key signature of one flat (B-flat). The tempo is marked 'Allegretto' and the time signature is 3/4. The score consists of two measures. The first measure contains a quarter note on G4, an eighth note on A4, and a sixteenth note on B4. The second measure contains a quarter note on C5, an eighth note on B4, and a sixteenth note on A4. The piece ends with a double bar line.

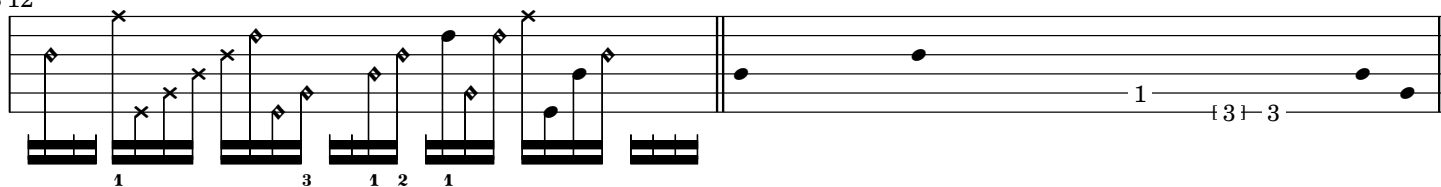
12'36"

12'48"

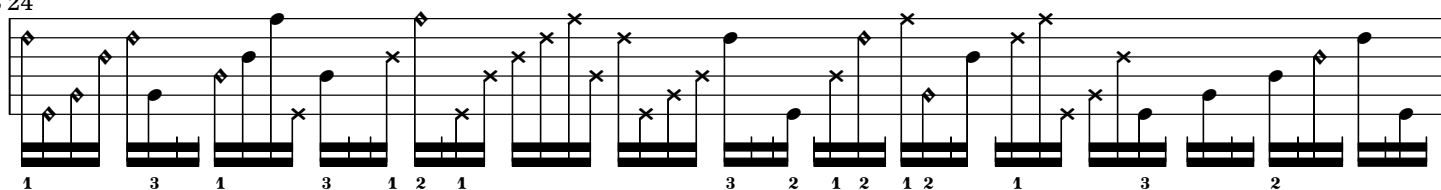
13'00"



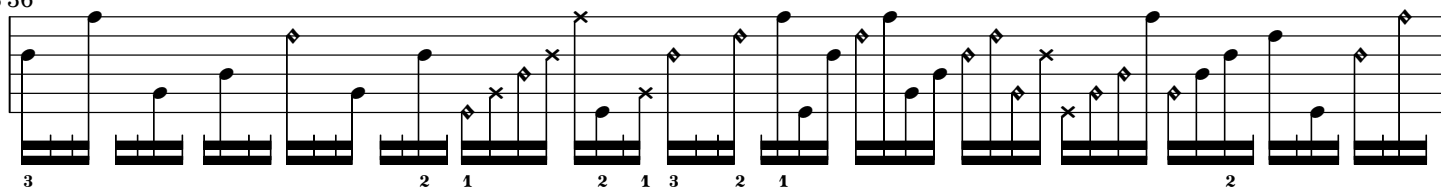
13'12"



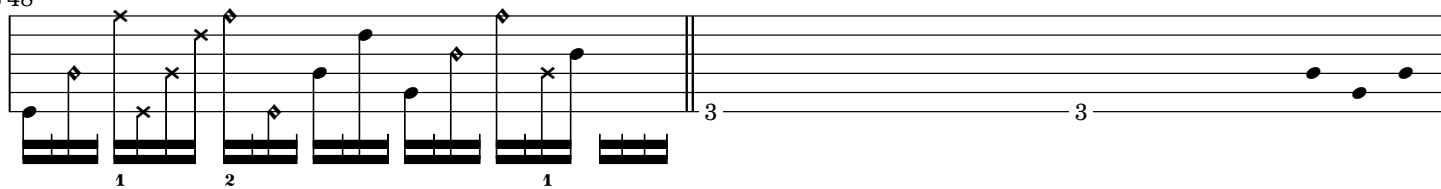
13'24"



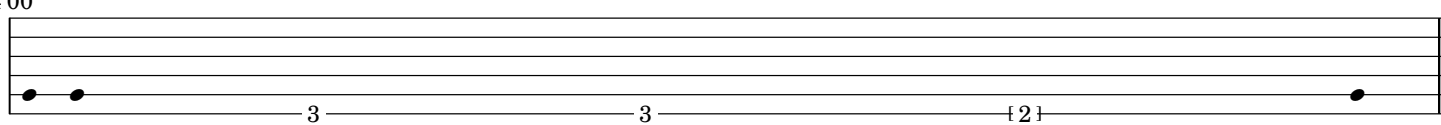
13'36"



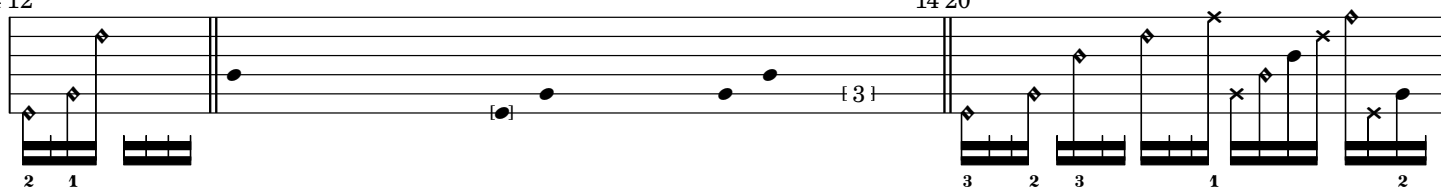
13'48"



14'00"

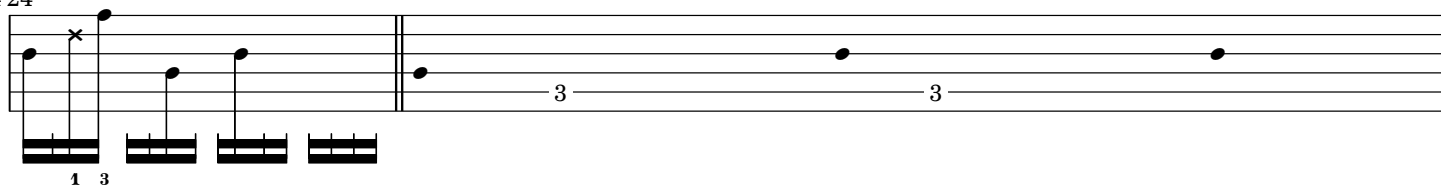


14'12"



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 then continues with a series of eighth notes and rests, with fingerings 3, 1, 2, 3, 1, 2 indicated below the notes.

14'44"

14'48"

Musical notation for 14'48". The staff shows a series of eighth notes and rests, with fingerings 1, 2, 1, 2, 3, 2, 3, 1, 2, 1, 2, 3, 2, 1 indicated below the notes.

15'00"

Musical notation for 15'00". The staff shows a series of eighth notes and rests, with fingerings 3, 2, 1, 2, 1, 2, 3, 1, 3, 2, 1, 3, 3, 2 indicated below the notes.

15'12"

Musical notation for 15'12". The staff shows a series of eighth notes and rests, with fingerings 3, 1, 2, 1, 3, 2, 3, 2 indicated below the notes. The notation ends with a double bar line and a rest of 1 measure, followed by a final note.

15'24"

Musical notation for 15'24". The staff shows a series of eighth notes and rests, with fingerings 2, 1, 2, 1, 2, 1, 3, 1, 2, 1, 3 indicated below the notes. The notation ends with a double bar line and a rest of 3 measures, followed by a final note.

15'36"

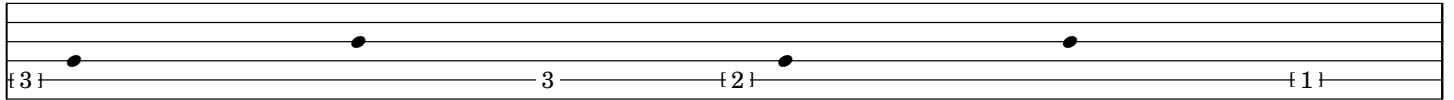
Musical notation for 15'36". The staff shows a long rest of 2 measures, followed by a double bar line and a rest of 2 measures, ending with a final note.

# *ostinato and interrupt*

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

percussion

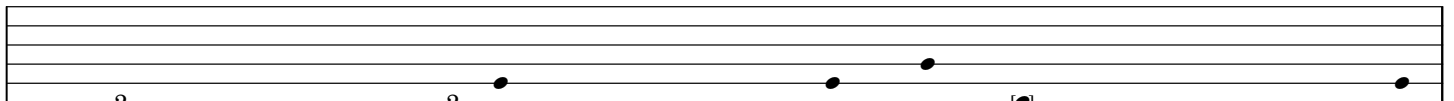
0'12"



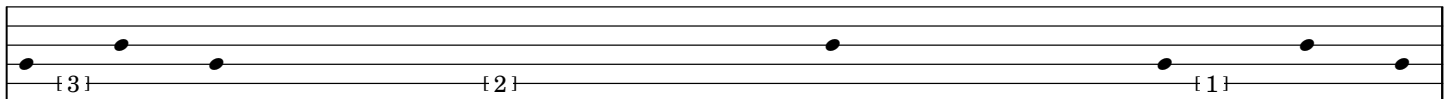
0'24"



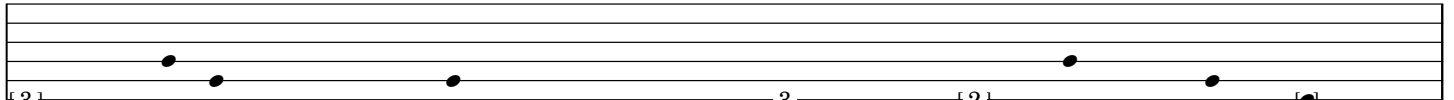
0'36"



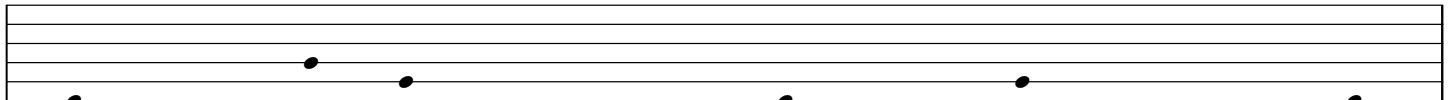
0'48"



1'00"

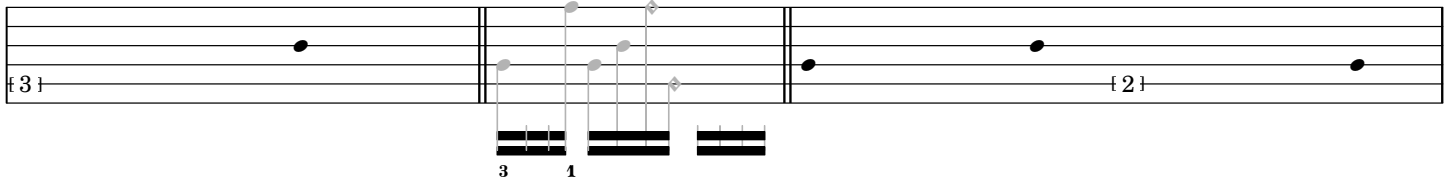


1'12"

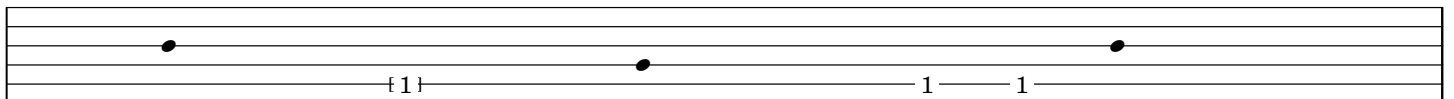


1'24"

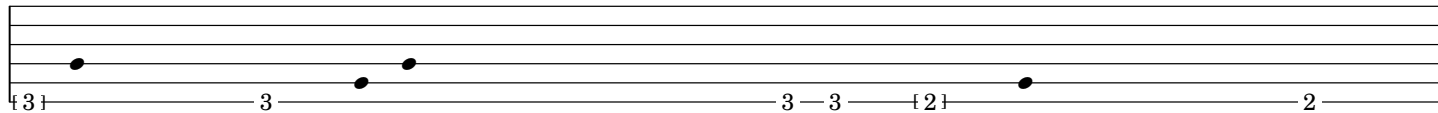
1'28"



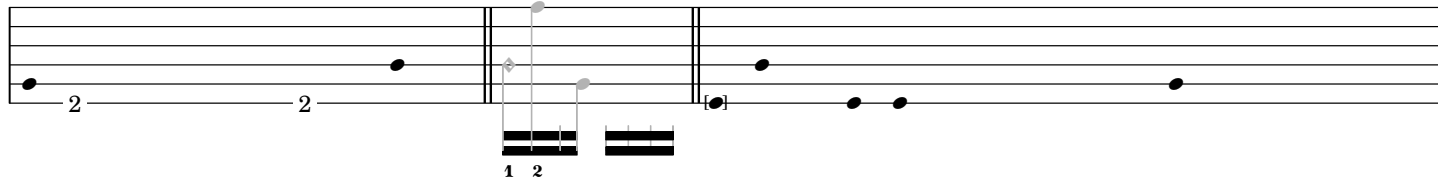
1'36"



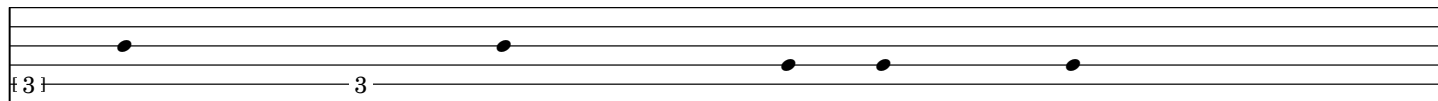
1'48"



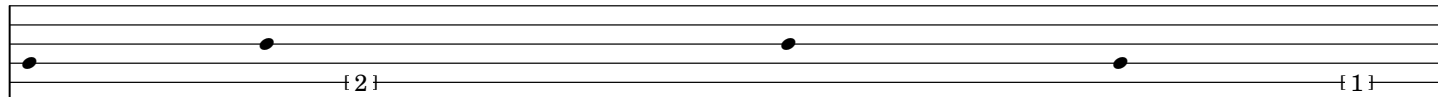
2'00"



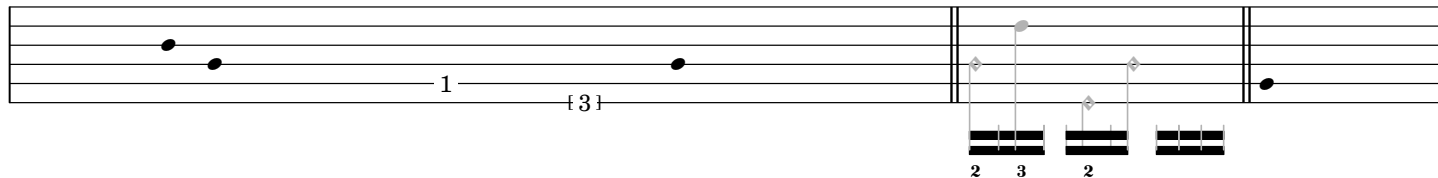
2'12"



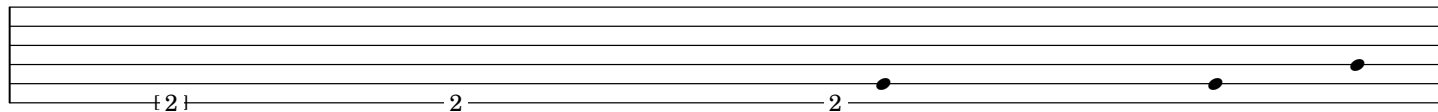
2'24"



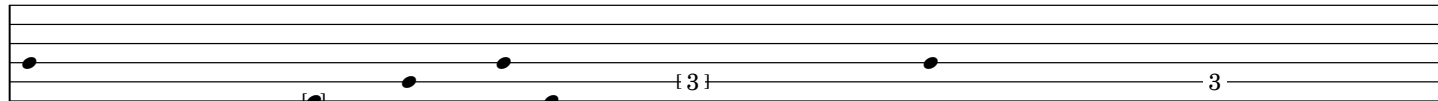
2'36"



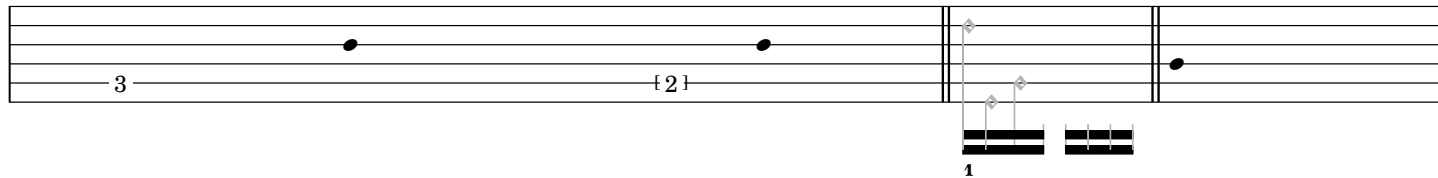
2'48"



3'00"

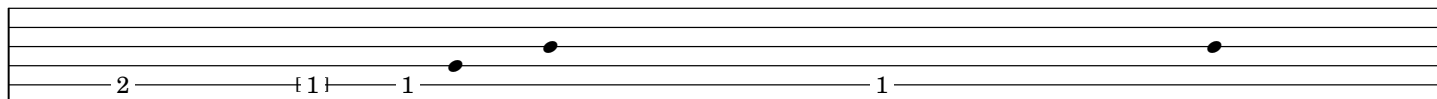


3'12"

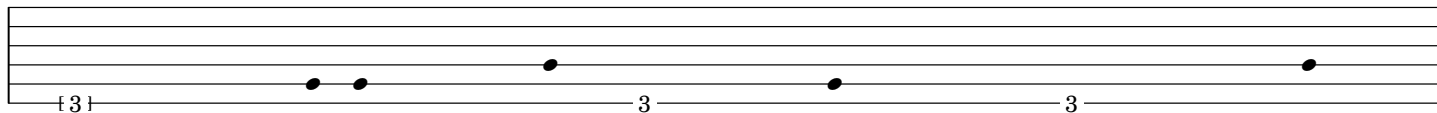




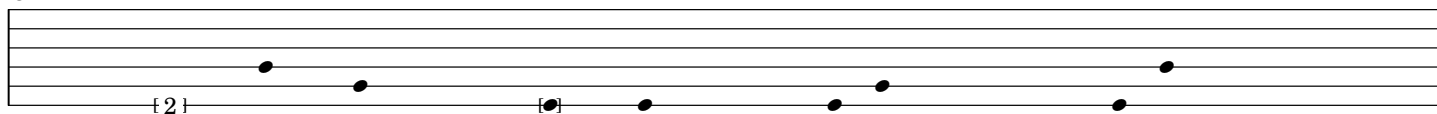
3'24"



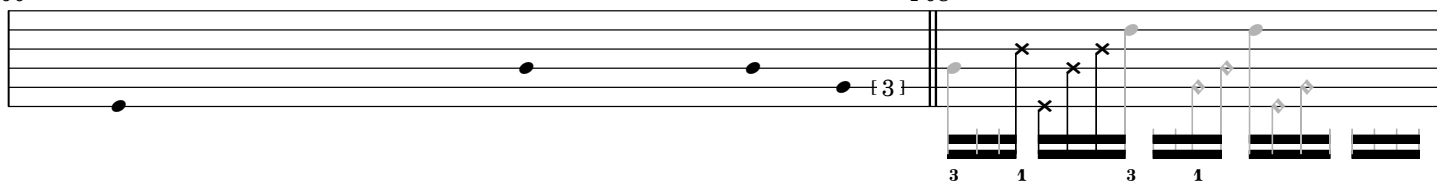
3'36"



3'48"

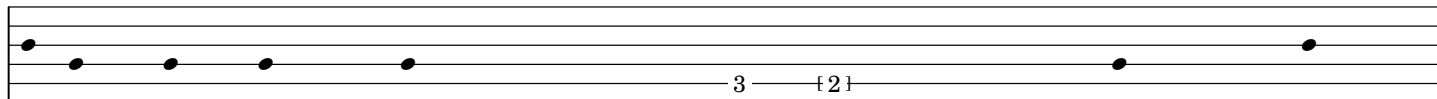


4'00"

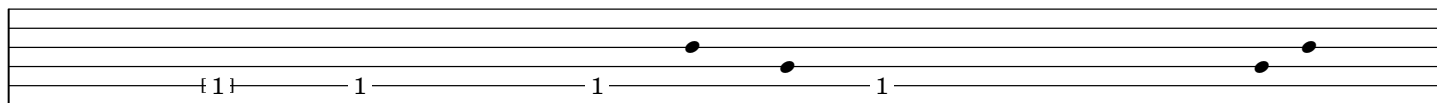


4'08"

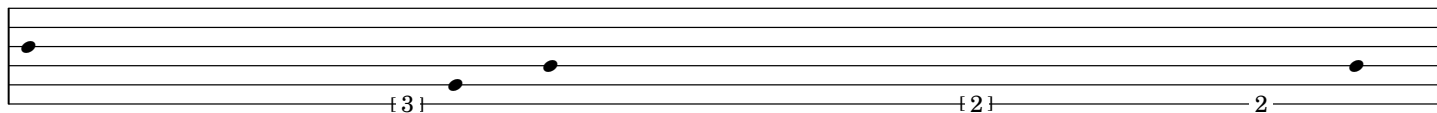
4'12"



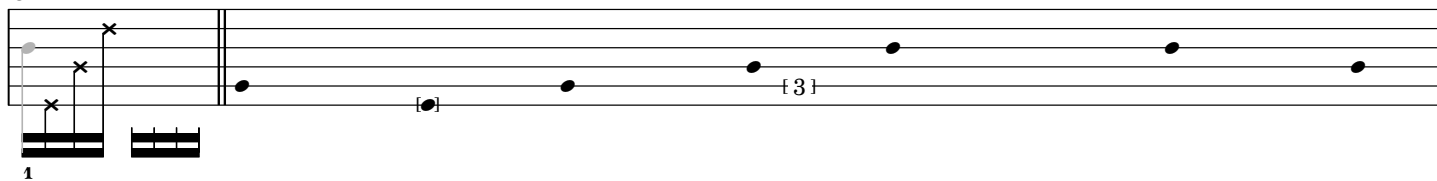
4'24"



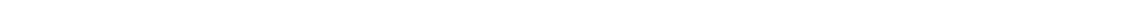
4'36"



4'48"



5'12"



2 2

5'36"

3

{ 2 }

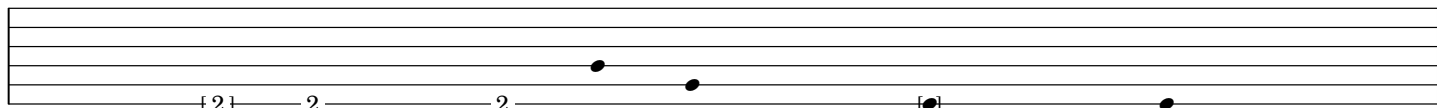
5'44"

1 2 1 3 1 3

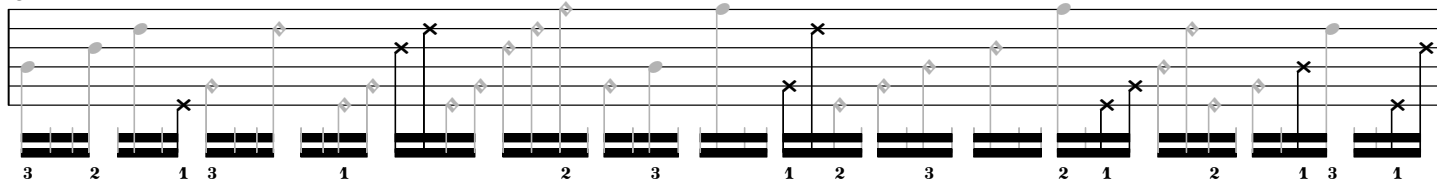
[illegible][illegible]

The musical score for 'The Great Wall of China' is presented on a single staff. It begins with a treble clef and a key signature of one sharp (F#). The tempo is marked 'Allegretto' and the time signature is 6/12. The score is divided into three measures by double bar lines. The first measure contains a whole note chord (F#4, A4, C5) and a fermata. The second measure contains a whole note chord (F#4, A4, C5) and a fermata. The third measure contains a whole note chord (F#4, A4, C5) and a fermata. The score is written for a single melodic line.

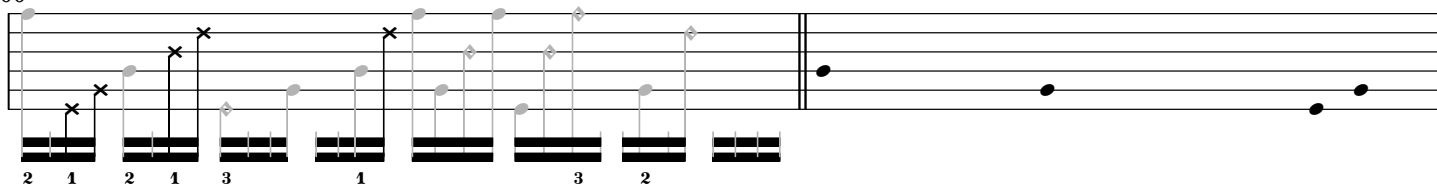
6'36"



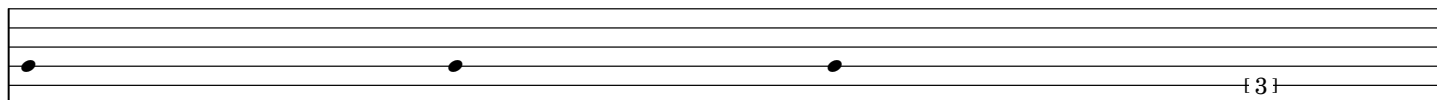
6'48"



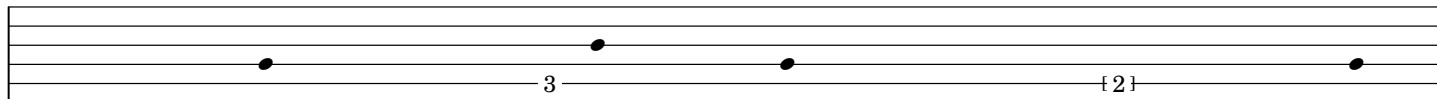
7'00"



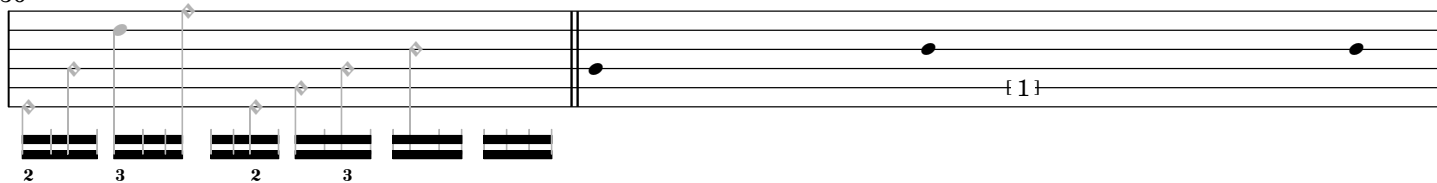
7'12"



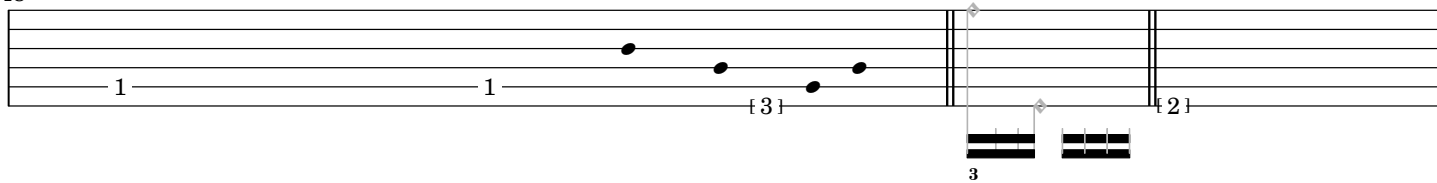
7'24"



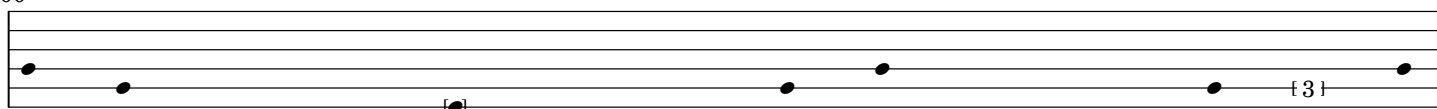
7'36"



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"

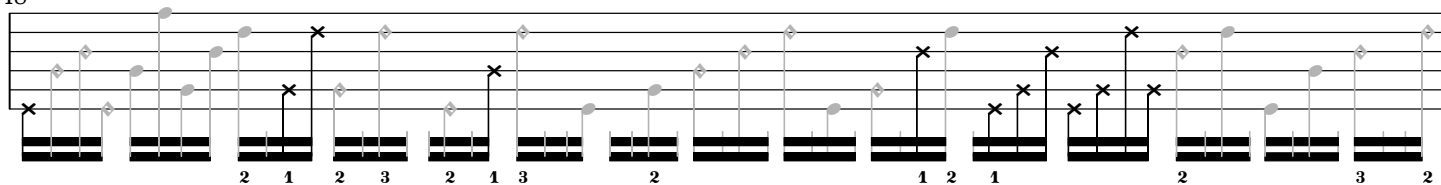
Musical notation for 9'24". The staff shows a sequence of notes.

9'36"

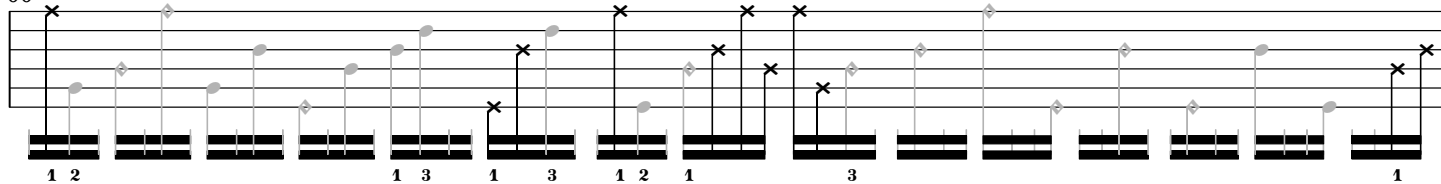
9'44"

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.

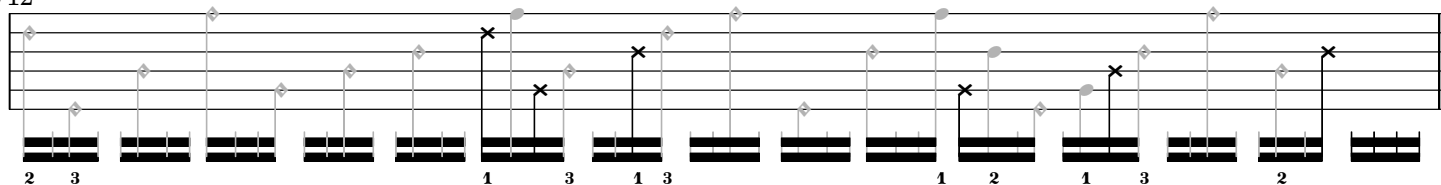
9'48"



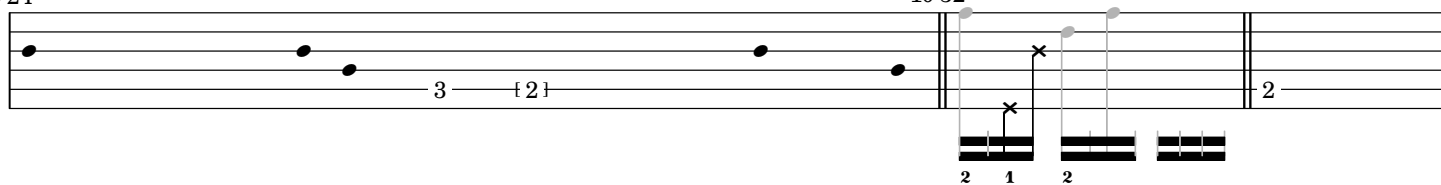
10'00"



10'12"

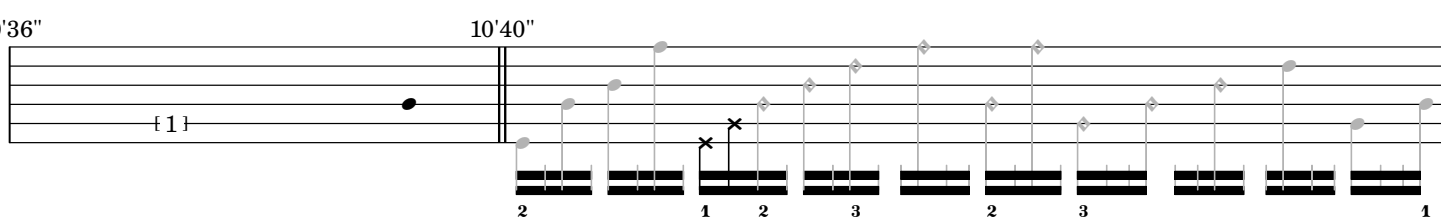


10'24"



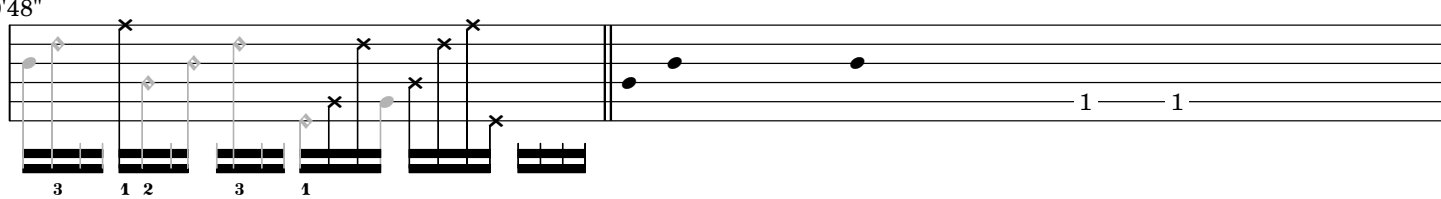
10'32"

10'36"

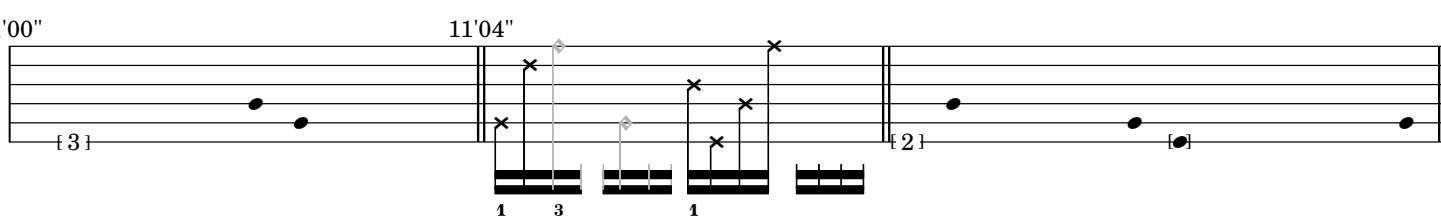


10'40"

10'48"

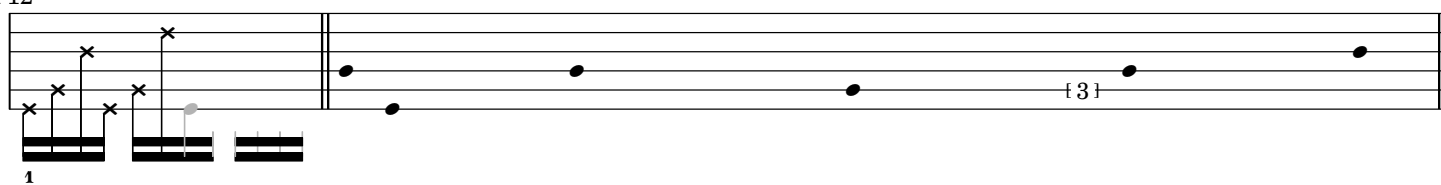


11'00"



11'04"

11'12"



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, and C5, with a final measure containing a whole note C5. 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, and C4, with a final measure containing a whole note C4. The piece is in 3/4 time, indicated by the '3' in the time signature. The first system ends with a double bar line.

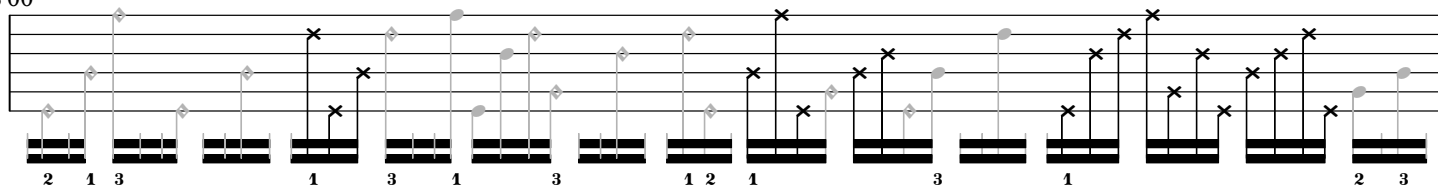
The first system of the musical score for 'The Rose Tree' is shown. It consists of a single staff with a treble clef and a key signature of one flat (B-flat). The melody begins with a quarter note on G4, followed by a quarter note on A4, and then a quarter note on B4. The lyrics 'The Rose Tree' are written below the staff. The first measure is marked with a '1' and a '3' below it, indicating a first and third ending. The first ending is marked with an 'X' and a '3' below it, indicating a first and third ending. The second ending is marked with a '1' and a '3' below it, indicating a first and third ending. The system ends with a double bar line and a repeat sign.

[illegible]

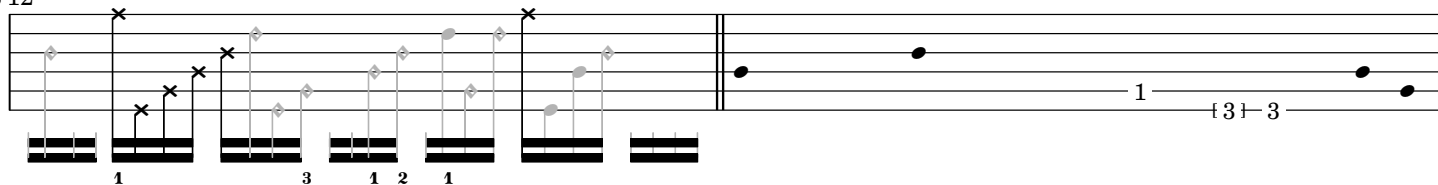
The first system of the musical score for 'The Rose Tree' consists of a treble clef, a key signature of one flat (B-flat), and a 2/4 time signature. The melody is written on a five-line staff. It begins with a quarter note on G4, followed by a quarter note on A4, then a quarter note on B4. This is followed by a quarter rest, then a quarter note on G4, then a quarter note on A4, and finally a quarter note on B4. The system ends with a double bar line. Below the staff, there are three groups of fingerings: '2' under the first G, '1' under the first A, and a group of '1' and '2' under the second G and A respectively.

The musical notation for the 'Piano' section is written on a grand staff (treble and bass clefs). It features a series of chords and single notes, many of which are marked with 'x' to indicate specific articulation or emphasis. The notation is organized into measures, with some measures containing multiple notes or chords. The overall style is minimalist and contemplative, typical of the 'Piano' section of the score.

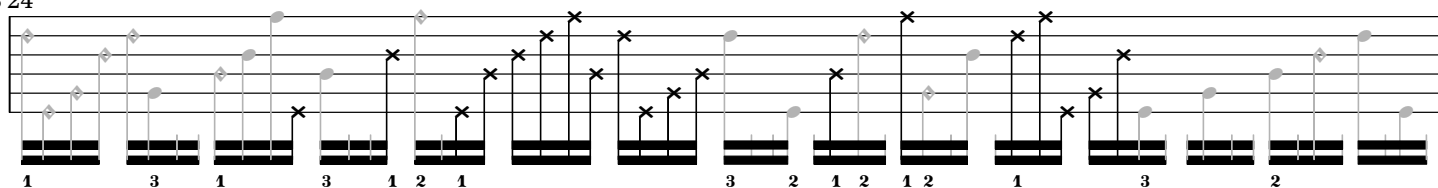
13'00"



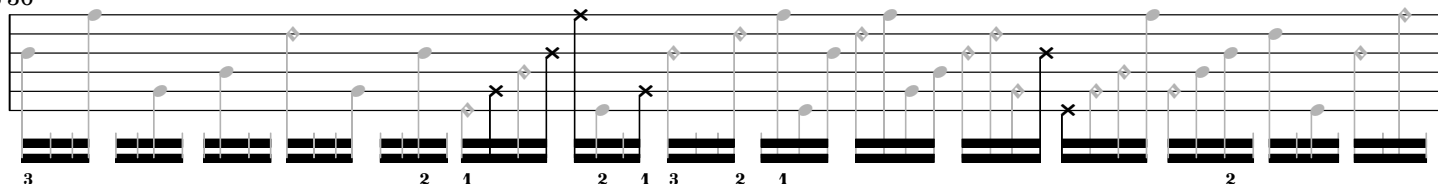
13'12"



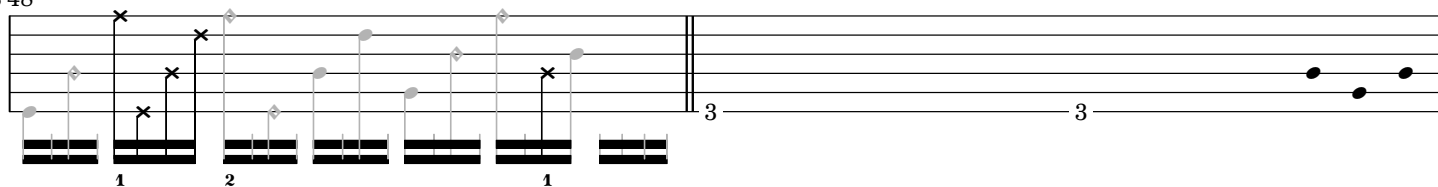
13'24"



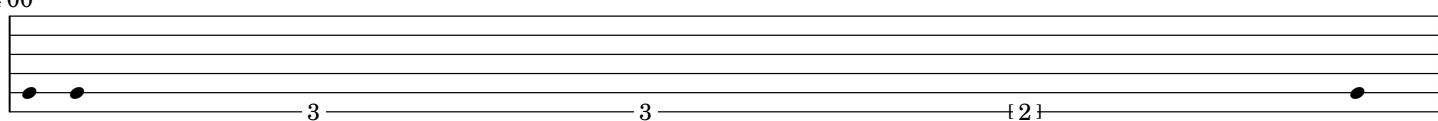
13'36"



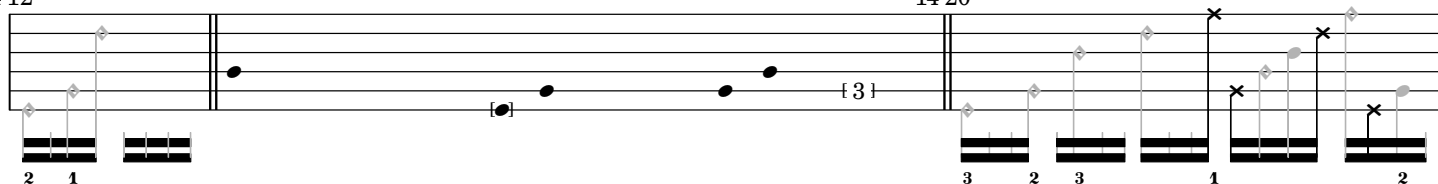
13'48"



14'00"

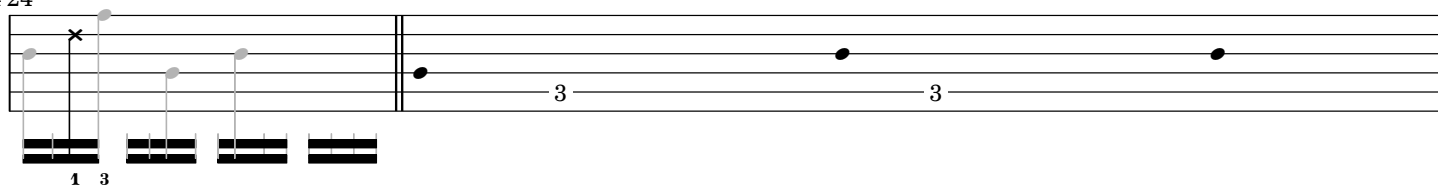


14'12"



14'20"

14'24"



14'36"

Musical notation for 14'36". The staff shows two whole notes, followed by a measure with a triplet of eighth notes (3) and a measure with a pair of eighth notes in brackets (2). A double bar line follows. The notation continues with eighth notes and sixteenth notes, including some marked with 'x'. Fingering numbers 3, 1, 2, 3, 1, 2 are shown below the notes.

14'44"

14'48"

Musical notation for 14'48". The staff shows a sequence of eighth and sixteenth notes, some marked with 'x'. Fingering numbers 1, 2, 1, 2, 3, 2, 3, 1, 2, 1, 2, 3, 2, 1 are shown below the notes.

15'00"

Musical notation for 15'00". The staff shows a sequence of eighth and sixteenth notes, some marked with 'x'. Fingering numbers 3, 2, 1, 2, 1, 2, 3, 1, 3, 2, 1, 3, 2 are shown below the notes.

15'12"

Musical notation for 15'12". The staff shows a sequence of eighth and sixteenth notes, some marked with 'x'. Fingering numbers 3, 1, 2, 1, 3, 2, 3, 2 are shown below the notes. A double bar line is followed by two whole notes and a measure with a pair of eighth notes in brackets (2) and a final measure with a single eighth note (1).

15'24"

Musical notation for 15'24". The staff shows a sequence of eighth and sixteenth notes, some marked with 'x'. Fingering numbers 2, 1, 2, 1, 2, 1, 3, 1, 2, 1 are shown below the notes. A double bar line is followed by two whole notes and a measure with a triplet of eighth notes in brackets (3).

15'36"

Musical notation for 15'36". The staff shows a sequence of whole notes, with a pair of eighth notes in brackets (2) and a final measure with a single eighth note (2) and a final measure with a single eighth note (2).

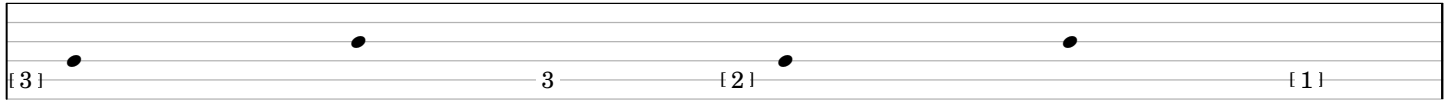


# *ostinato and interrupt*

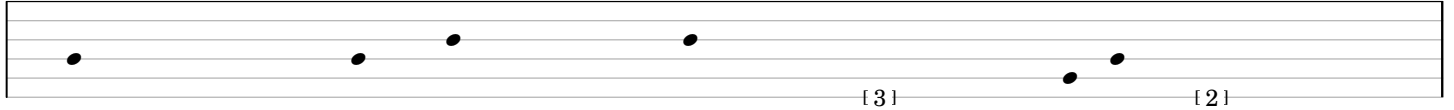
ensemble part 1

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

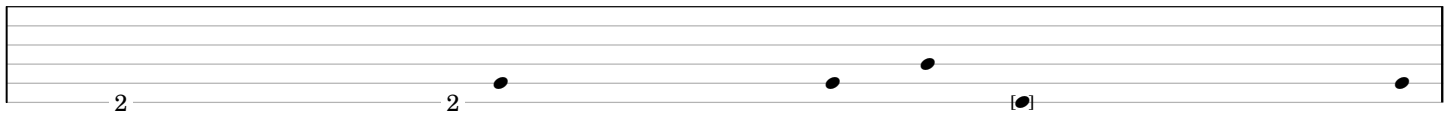
0'12"



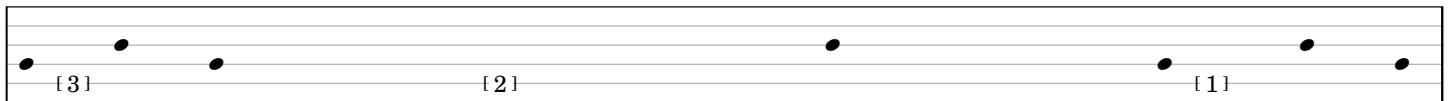
0'24"



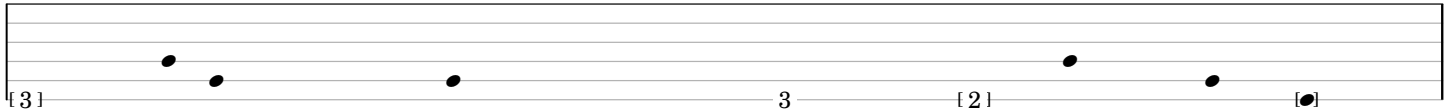
0'36"



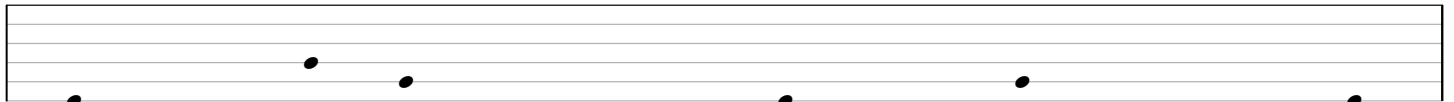
0'48"



1'00"

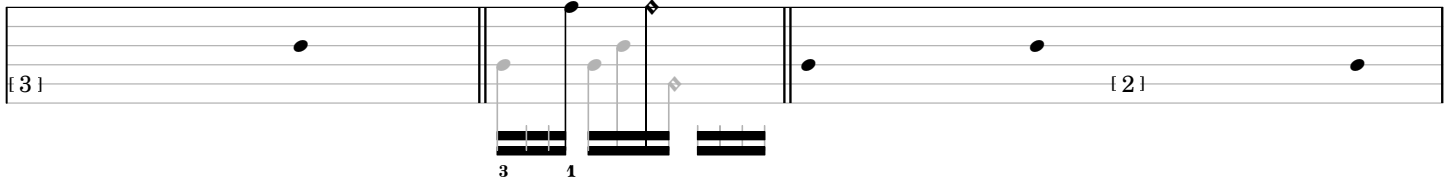


1'12"

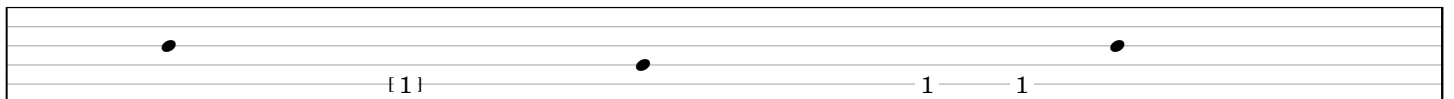


1'24"

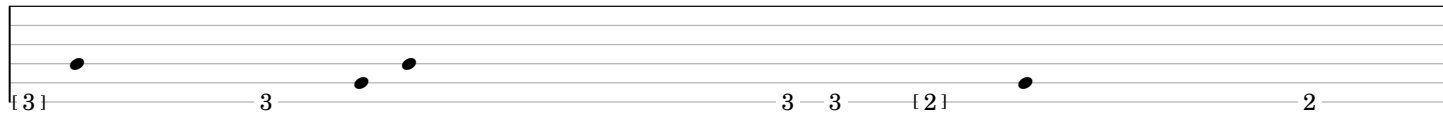
1'28"



1'36"

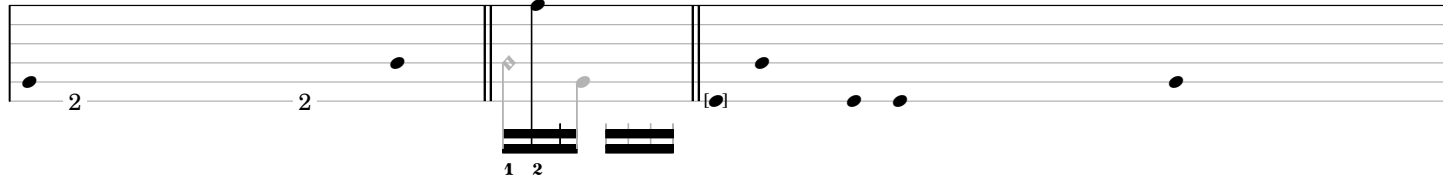


1'48"

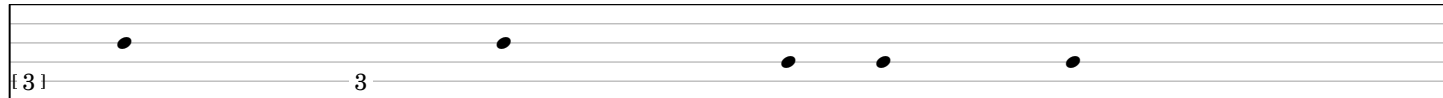


2'00"

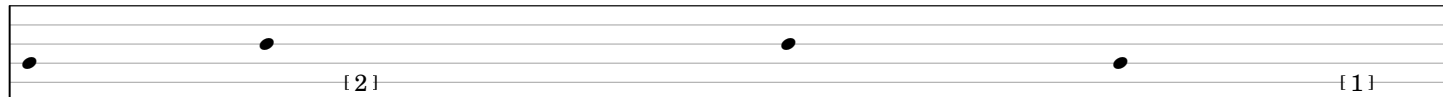
2'04"



2'12"

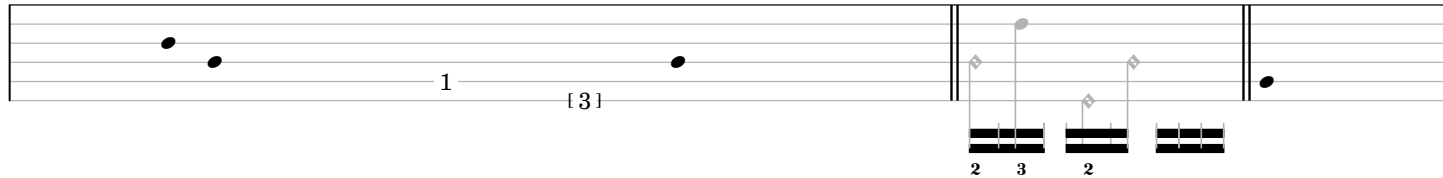


2'24"



2'36"

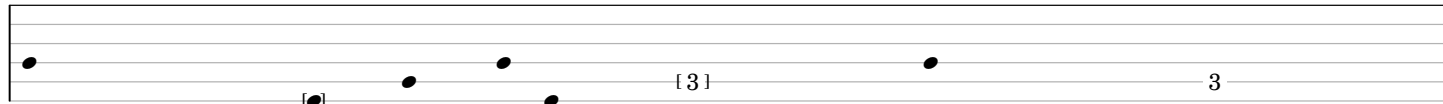
2'44"



2'48"

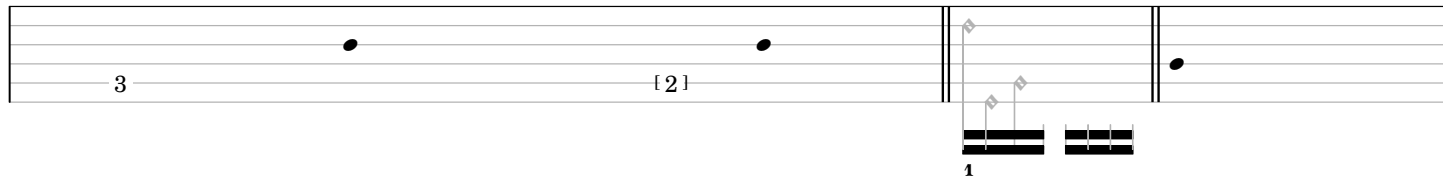


3'00"



3'12"

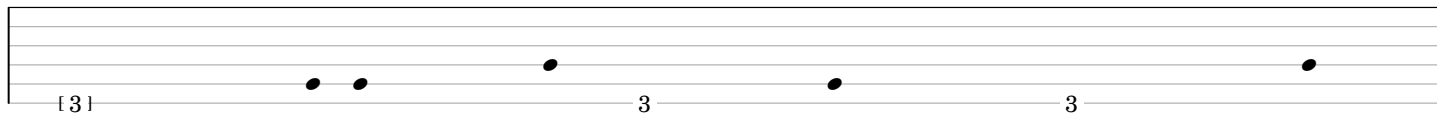
3'20"



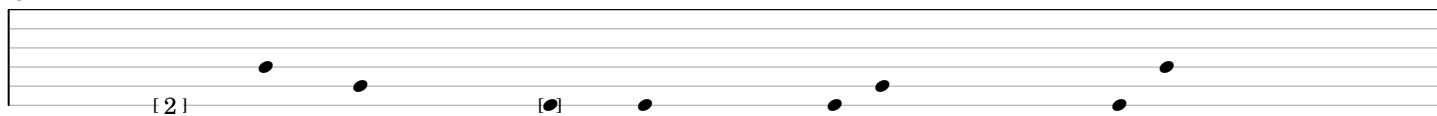
3'24"



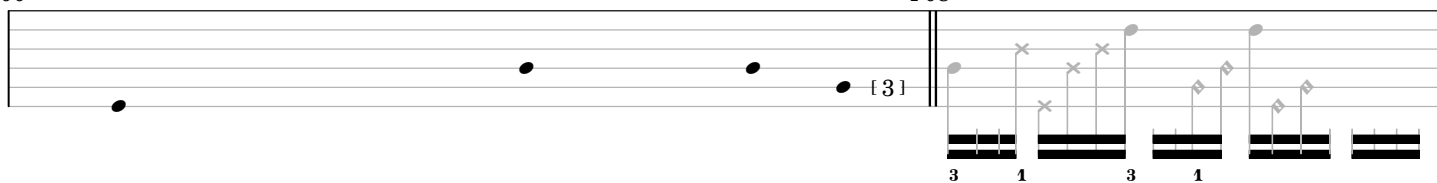
3'36"



3'48"

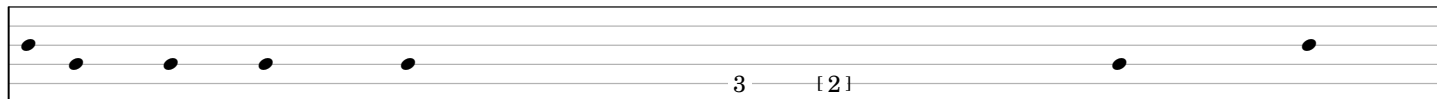


4'00"



4'08"

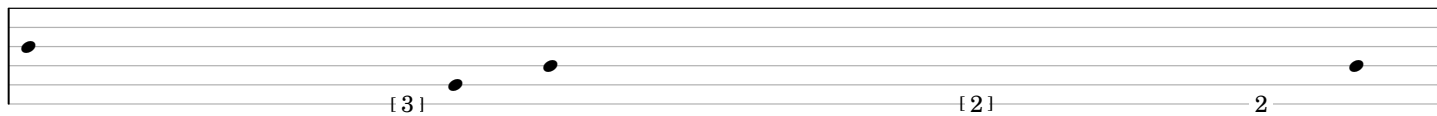
4'12"



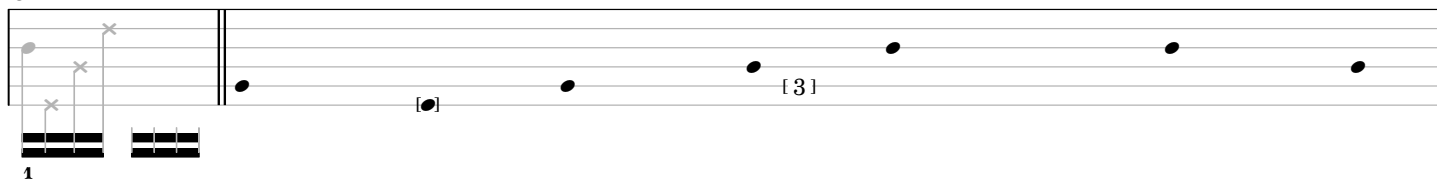
4'24"



4'36"



4'48"



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.

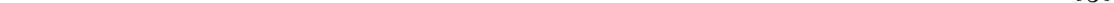
The first system of the musical score for 'The Little Boat' consists of a single staff. It begins with a treble clef and a key signature of one flat (B-flat). The melody starts with a quarter note G4, followed by a quarter note A4, and then a quarter note B4. This is followed by a half note G4, and then a half note F4. The melody ends with a quarter note E4. The staff is marked with a '3' below the first measure and a '[ 2 ]' below the fifth measure. The system 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, a half note E4, and a half note D4. 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, a half note E3, and a half note D3. 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), Bb4 (quarter), A4 (quarter), G4 (quarter), F4 (quarter), E4 (quarter), D4 (half). The second line is: C4 (half), Bb3 (quarter), A3 (quarter), G3 (quarter), F3 (quarter), E3 (quarter), D3 (half). The third line is: C3 (half), Bb2 (quarter), A2 (quarter), G2 (quarter), F2 (quarter), E2 (quarter), D2 (half). The fourth line is: C2 (half), Bb1 (quarter), A1 (quarter), G1 (quarter), F1 (quarter), E1 (quarter), D1 (half). The fifth line is: C1 (half), Bb0 (quarter), A0 (quarter), G0 (quarter), F0 (quarter), E0 (quarter), D0 (half). The sixth line is: C0 (half), Bb-1 (quarter), A-1 (quarter), G-1 (quarter), F-1 (quarter), E-1 (quarter), D-1 (half). The seventh line is: C-1 (half), Bb-2 (quarter), A-2 (quarter), G-2 (quarter), F-2 (quarter), E-2 (quarter), D-2 (half). The eighth line is: C-2 (half), Bb-3 (quarter), A-3 (quarter), G-3 (quarter), F-3 (quarter), E-3 (quarter), D-3 (half). The ninth line is: C-3 (half), Bb-4 (quarter), A-4 (quarter), G-4 (quarter), F-4 (quarter), E-4 (quarter), D-4 (half). The tenth line is: C-4 (half), Bb-5 (quarter), A-5 (quarter), G-5 (quarter), F-5 (quarter), E-5 (quarter), D-5 (half). The eleventh line is: C-5 (half), Bb-6 (quarter), A-6 (quarter), G-6 (quarter), F-6 (quarter), E-6 (quarter), D-6 (half). The twelfth line is: C-6 (half), Bb-7 (quarter), A-7 (quarter), G-7 (quarter), F-7 (quarter), E-7 (quarter), D-7 (half). The thirteenth line is: C-7 (half), Bb-8 (quarter), A-8 (quarter), G-8 (quarter), F-8 (quarter), E-8 (quarter), D-8 (half). The fourteenth line is: C-8 (half), Bb-9 (quarter), A-9 (quarter), G-9 (quarter), F-9 (quarter), E-9 (quarter), D-9 (half). The fifteenth line is: C-9 (half), Bb-10 (quarter), A-10 (quarter), G-10 (quarter), F-10 (quarter), E-10 (quarter), D-10 (half). The sixteenth line is: C-10 (half), Bb-11 (quarter), A-11 (quarter), G-11 (quarter), F-11 (quarter), E-11 (quarter), D-11 (half). The seventeenth line is: C-11 (half), Bb-12 (quarter), A-12 (quarter), G-12 (quarter), F-12 (quarter), E-12 (quarter), D-12 (half). The eighteenth line is: C-12 (half), Bb-13 (quarter), A-13 (quarter), G-13 (quarter), F-13 (quarter), E-13 (quarter), D-13 (half). The nineteenth line is: C-13 (half), Bb-14 (quarter), A-14 (quarter), G-14 (quarter), F-14 (quarter), E-14 (quarter), D-14 (half). The twentieth line is: C-14 (half), Bb-15 (quarter), A-15 (quarter), G-15 (quarter), F-15 (quarter), E-15 (quarter), D-15 (half). The twenty-first line is: C-15 (half), Bb-16 (quarter), A-16 (quarter), G-16 (quarter), F-16 (quarter), E-16 (quarter), D-16 (half). The twenty-second line is: C-16 (half), Bb-17 (quarter), A-17 (quarter), G-17 (quarter), F-17 (quarter), E-17 (quarter), D-17 (half). The twenty-third line is: C-17 (half), Bb-18 (quarter), A-18 (quarter), G-18 (quarter), F-18 (quarter), E-18 (quarter), D-18 (half). The twenty-fourth line is: C-18 (half), Bb-19 (quarter), A-19 (quarter), G-19 (quarter), F-19 (quarter), E-19 (quarter), D-19 (half). The twenty-fifth line is: C-19 (half), Bb-20 (quarter), A-20 (quarter), G-20 (quarter), F-20 (quarter), E-20 (quarter), D-20 (half). The twenty-sixth line is: C-20 (half), Bb-21 (quarter), A-21 (quarter), G-21 (quarter), F-21 (quarter), E-21 (quarter), D-21 (half). The twenty-seventh line is: C-21 (half), Bb-22 (quarter), A-22 (quarter), G-22 (quarter), F-22 (quarter), E-22 (quarter), D-22 (half). The twenty-eighth line is: C-22 (half), Bb-23 (quarter), A-23 (quarter), G-23 (quarter), F-23 (quarter), E-23 (quarter), D-23 (half). The twenty-ninth line is: C-23 (half), Bb-24 (quarter), A-24 (quarter), G-24 (quarter), F-24 (quarter), E-24 (quarter), D-24 (half). The thirtieth line is: C-24 (half), Bb-25 (quarter), A-25 (quarter), G-25 (quarter), F-25 (quarter), E-25 (quarter), D-25 (half). The thirty-first line is: C-25 (half), Bb-26 (quarter), A-26 (quarter), G-26 (quarter), F-26 (quarter), E-26 (quarter), D-26 (half). The thirty-second line is: C-26 (half), Bb-27 (quarter), A-27 (quarter), G-27 (quarter), F-27 (quarter), E-27 (quarter), D-27 (half). The thirty-third line is: C-27 (half), Bb-28 (quarter), A-28 (quarter), G-28 (quarter), F-28 (quarter), E-28 (quarter), D-28 (half). The thirty-fourth line is: C-28 (half), Bb-29 (quarter), A-29 (quarter), G-29 (quarter), F-29 (quarter), E-29 (quarter), D-29 (half). The thirty-fifth line is: C-29 (half), Bb-30 (quarter), A-30 (quarter), G-30 (quarter), F-30 (quarter), E-30 (quarter), D-30 (half). The thirty-sixth line is: C-30 (half), Bb-31 (quarter), A-31 (quarter), G-31 (quarter), F-31 (quarter), E-31 (quarter), D-31 (half). The thirty-seventh line is: C-31 (half), Bb-32 (quarter), A-32 (quarter), G-32 (quarter), F-32 (quarter), E-32 (quarter), D-32 (half). The thirty-eighth line is: C-32 (half), Bb-33 (quarter), A-33 (quarter), G-33 (quarter), F-33 (quarter), E-33 (quarter), D-33 (half). The thirty-ninth line is: C-33 (half), Bb-34 (quarter), A-34 (quarter), G-34 (quarter), F-34 (quarter), E-34 (quarter), D-34 (half). The fortieth line is: C-34 (half), Bb-35 (quarter), A-35 (quarter), G-35 (quarter), F-35 (quarter), E-35 (quarter), D-35 (half). The forty-first line is: C-35 (half), Bb-36 (quarter), A-36 (quarter), G-36 (quarter), F-36 (quarter), E-36 (quarter), D-36 (half). The forty-second line is: C-36 (half), Bb-37 (quarter), A-37 (quarter), G-37 (quarter), F-37 (quarter), E-37 (quarter), D-37 (half). The forty-third line is: C-37 (half), Bb-38 (quarter), A-38 (quarter), G-38 (quarter), F-38 (quarter), E-38 (quarter), D-38 (half). The forty-fourth line is: C-38 (half), Bb-39 (quarter), A-39 (quarter), G-39 (quarter), F-39 (quarter), E-39 (quarter), D-39 (half). The forty-fifth line is: C-39 (half), Bb-40 (quarter), A-40 (quarter), G-40 (quarter), F-40 (quarter), E-40 (quarter), D-40 (half). The forty-sixth line is: C-40 (half), Bb-41 (quarter), A-41 (quarter), G-41 (quarter), F-41 (quarter), E-41 (quarter), D-41 (half). The forty-seventh line is: C-41 (half), Bb-42 (quarter), A-42 (quarter), G-42 (quarter), F-42 (quarter), E-42 (quarter), D-42 (half). The forty-eighth line is: C-42 (half), Bb-43 (quarter), A-43 (quarter), G-43 (quarter), F-43 (quarter), E-43 (quarter), D-43 (half). The forty-ninth line is: C-43 (half), Bb-44 (quarter), A-44 (quarter), G-44 (quarter), F-44 (quarter), E-44 (quarter), D-44 (half). The fiftieth line is: C-44 (half), Bb-45 (quarter), A-45 (quarter), G-45 (quarter), F-45 (quarter), E-45 (quarter), D-45 (half). The fifty-first line is: C-45 (half), Bb-46 (quarter), A-46 (quarter), G-46 (quarter), F-46 (quarter), E-46 (quarter), D-46 (half). The fifty-second line is: C-46 (half), Bb-47 (quarter), A-47 (quarter), G-47 (quarter), F-47 (quarter), E-47 (quarter), D-47 (half). The fifty-third line is: C-47 (half), Bb-48 (quarter), A-48 (quarter), G-48 (quarter), F-48 (quarter), E-48 (quarter), D-48 (half). The fifty-fourth line is: C-48 (half), Bb-49 (quarter), A-49 (quarter), G-49 (quarter), F-49 (quarter), E-49 (quarter), D-49 (half). The fifty-fifth line is: C-49 (half), Bb-50 (quarter), A-50 (quarter), G-50 (quarter), F-50 (quarter), E-50 (quarter), D-50 (half). The fifty-sixth line is: C-50 (half), Bb-51 (quarter), A-51 (quarter), G-51 (quarter), F-51 (quarter), E-51 (quarter), D-51 (half). The fifty-seventh line is: C-51 (half), Bb-52 (quarter), A-52 (quarter), G-52 (quarter), F-52 (quarter), E-52 (quarter), D-52 (half). The fifty-eighth line is: C-52 (half), Bb-53 (quarter), A-53 (quarter), G-53 (quarter), F-53 (quarter), E-53 (quarter), D-53 (half). The fifty-ninth line is: C-53 (half), Bb-54 (quarter), A-54 (quarter), G-54 (quarter), F-54 (quarter), E-54 (quarter), D-54 (half). The sixtieth line is: C-54 (half), Bb-55 (quarter), A-55 (quarter), G-55 (quarter), F-55 (quarter), E-55 (quarter), D-55 (half). The sixty-first line is: C-55 (half), Bb-56 (quarter), A-56 (quarter), G-56 (quarter), F-56 (quarter), E-56 (quarter), D-56 (half). The sixty-second line is: C-56 (half), Bb-57 (quarter), A-57 (quarter), G-57 (quarter), F-57 (quarter), E-57 (quarter), D-57 (half). The sixty-third line is: C-57 (half), Bb-58 (quarter), A-58 (quarter), G-58 (quarter), F-58 (quarter), E-58 (quarter), D-58 (half). The sixty-fourth line is: C-58 (half), Bb-59 (quarter), A-59 (quarter), G-59 (quarter), F-59 (quarter), E-59 (quarter), D-59 (half). The sixty-fifth line is: C-59 (half), Bb-60 (quarter), A-60 (quarter), G-60 (quarter), F-60 (quarter), E-60 (quarter), D-60 (half). The sixty-sixth line is: C-60 (half), Bb-61 (quarter), A-61 (quarter), G-61 (quarter), F-61 (quarter), E-61 (quarter), D-61 (half). The sixty-seventh line is: C-61 (half), Bb-62 (quarter), A-62 (quarter), G-62 (quarter), F-62 (quarter), E-62 (quarter), D-62 (half). The sixty-eighth line is: C-62 (half), Bb-63 (quarter), A-63 (quarter), G-63 (quarter), F-63 (quarter), E-63 (quarter), D-63 (half). The sixty-ninth line is: C-63 (half), Bb-64 (quarter), A-64 (quarter), G-64 (quarter), F-64 (quarter), E-64 (quarter), D-64 (half). The seventieth line is: C-64 (half), Bb-65 (quarter), A-65 (quarter), G-65 (quarter), F-65 (quarter), E-65 (quarter), D-65 (half). The seventy-first line is: C-65 (half), Bb-66 (quarter), A-66 (quarter), G-66 (quarter), F-66 (quarter), E-66 (quarter), D-66 (half). The seventy-second line is: C-66 (half), Bb-67 (quarter), A-67 (quarter), G-67 (quarter), F-67 (quarter), E-67 (quarter), D-67 (half). The seventy-third line is: C-67 (half), Bb-68 (quarter), A-68 (quarter), G-68 (quarter), F-68 (quarter), E-68 (quarter), D-68 (half). The seventy-fourth line is: C-68 (half), Bb-69 (quarter), A-69 (quarter), G-69 (quarter), F-69 (quarter), E-69 (quarter

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  $x=3$ .



A horizontal bar with four dots. Below the second dot from the left is the label '3'. Below the third dot from the left is the label '[2]'.

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"

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"

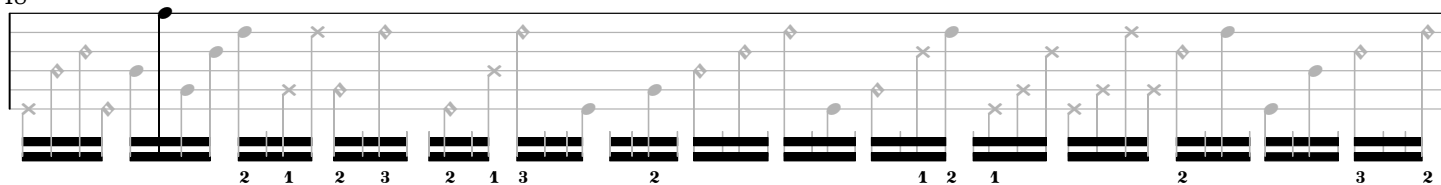
Musical notation for 9'24". The staff shows a sequence of notes.

9'36"

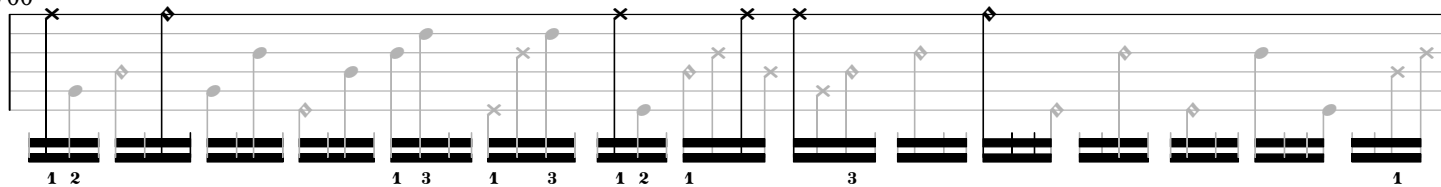
9'44"

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.

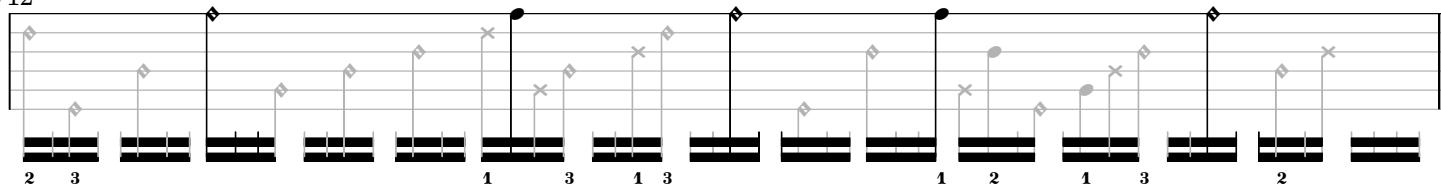
9'48"



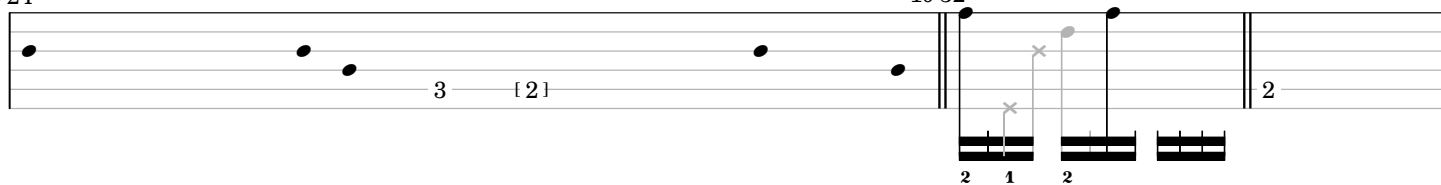
10'00"



10'12"

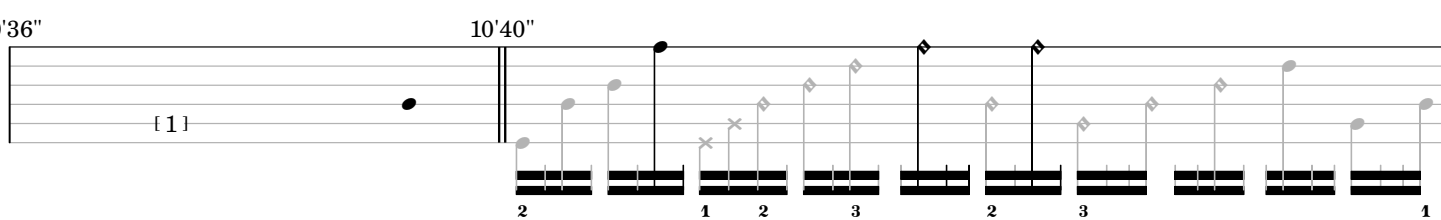


10'24"



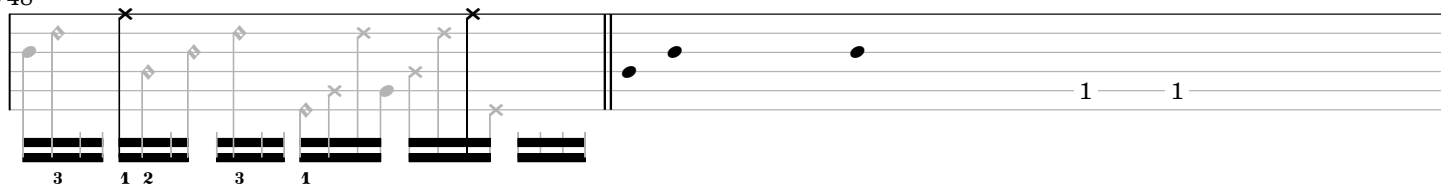
10'32"

10'36"

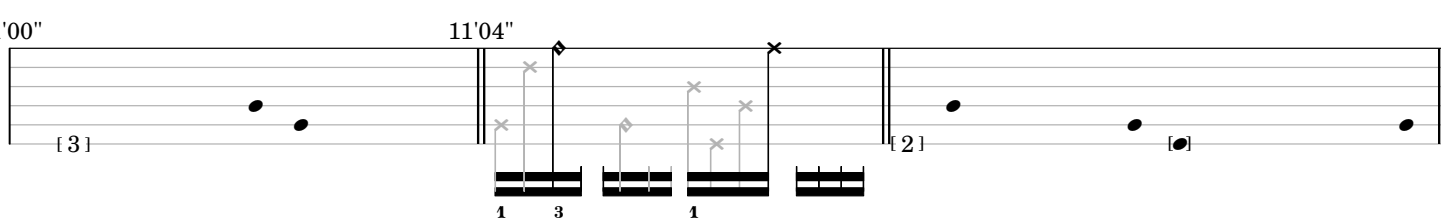


10'40"

10'48"

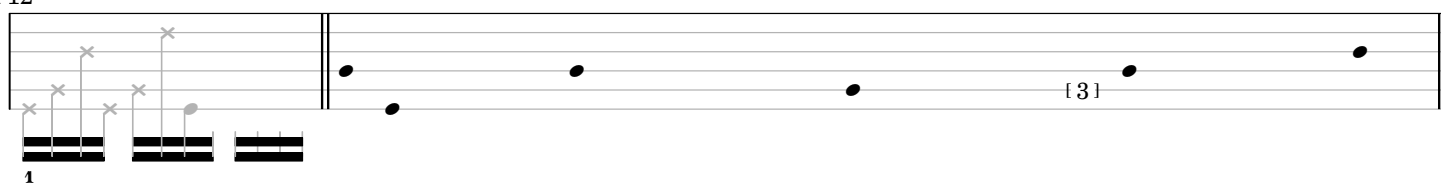


11'00"



11'04"

11'12"



11'24"

Musical notation for 11'24". The staff shows a sequence of notes with various ornaments (diamonds, crosses) and fingerings (2, 1, 3, 2, 3, 2, 1, 3, 1, 2, 1). The notation includes a double bar line and a repeat sign.

11'36"

Musical notation for 11'36". The staff shows a sequence of notes with various ornaments (diamonds, crosses) and fingerings (1, 2, 1). The notation includes a double bar line and a repeat sign.

11'44"

11'48"

Musical notation for 11'48". The staff shows a sequence of notes with various ornaments (diamonds, crosses) and fingerings (3, [2]). The notation includes a double bar line and a repeat sign.

12'00"

Musical notation for 12'00". The staff shows a sequence of notes with various ornaments (diamonds, crosses) and fingerings (1, 2, 1, 2, 1, 2, 1, 2, 2, 1, 2, 1). The notation includes a double bar line and a repeat sign.

12'12"

Musical notation for 12'12". The staff shows a sequence of notes with various ornaments (diamonds, crosses) and fingerings ([3]). The notation includes a double bar line and a repeat sign.

12'24"

Musical notation for 12'24". The staff shows a sequence of notes with various ornaments (diamonds, crosses) and fingerings (2, 1). The notation includes a double bar line and a repeat sign.

12'36"

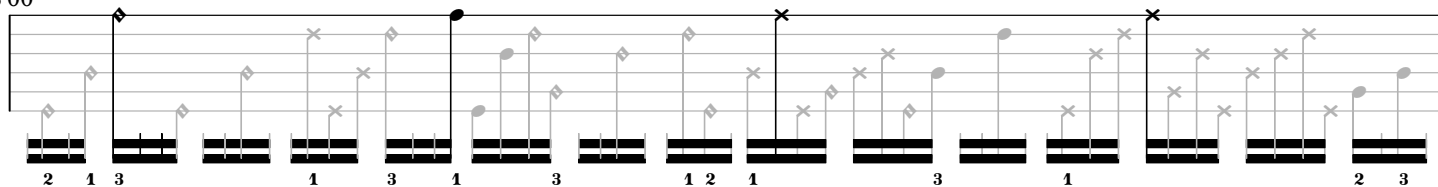
Musical notation for 12'36". The staff shows a sequence of notes with various ornaments (diamonds, crosses) and fingerings (3, 1, 2, [1], 1). The notation includes a double bar line and a repeat sign.

12'48"

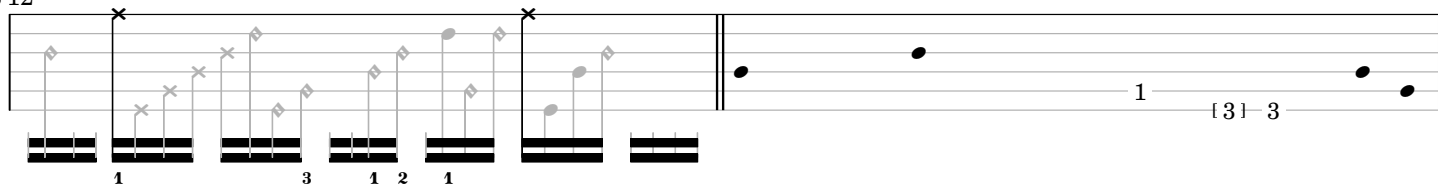
Musical notation for 12'48". The staff shows a sequence of notes with various ornaments (diamonds, crosses) and fingerings (2, 3, 1, 2, 1, 3, 1, 3, 1, 3, 1, 3). The notation includes a double bar line and a repeat sign.



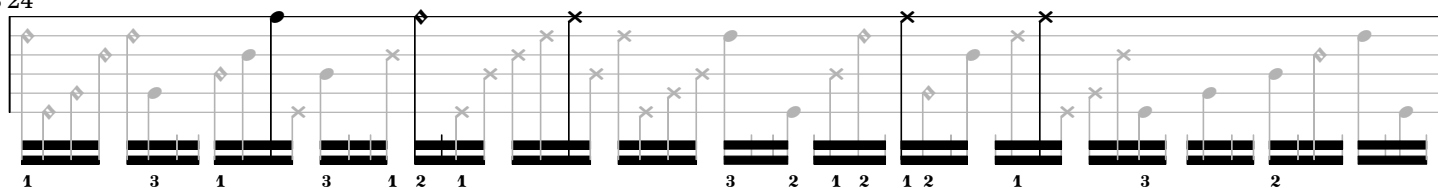
13'00"



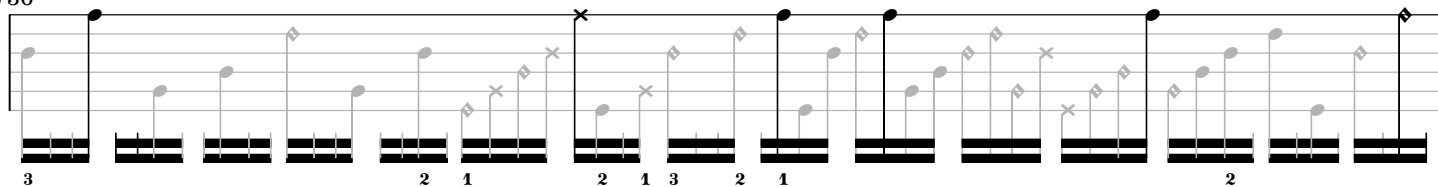
13'12"



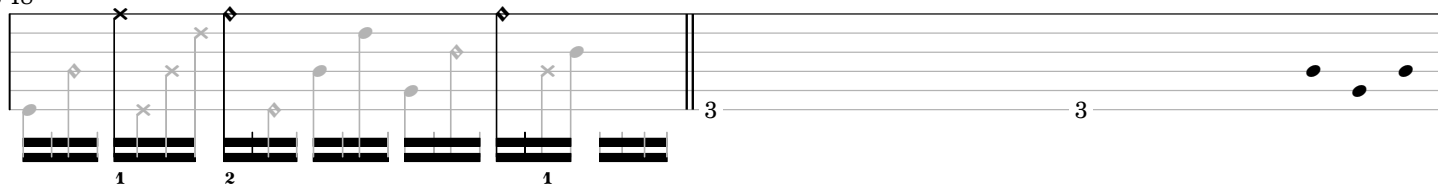
13'24"



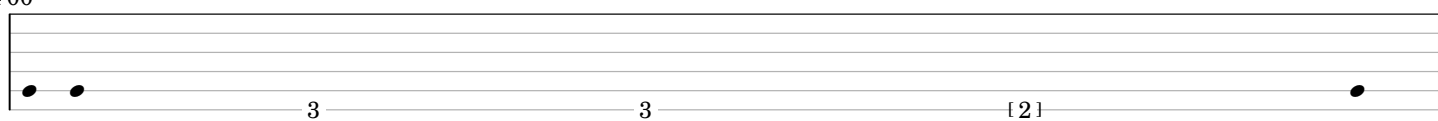
13'36"



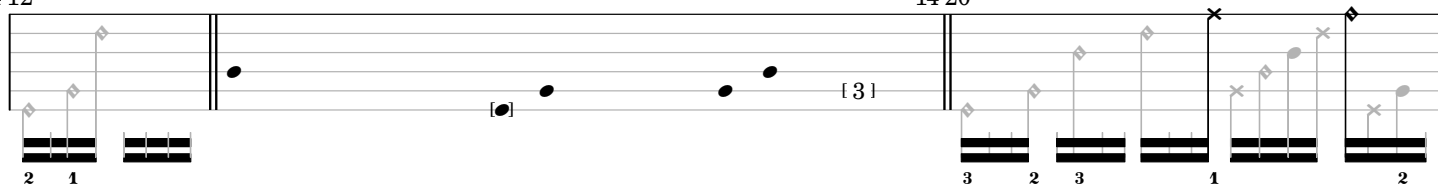
13'48"



14'00"

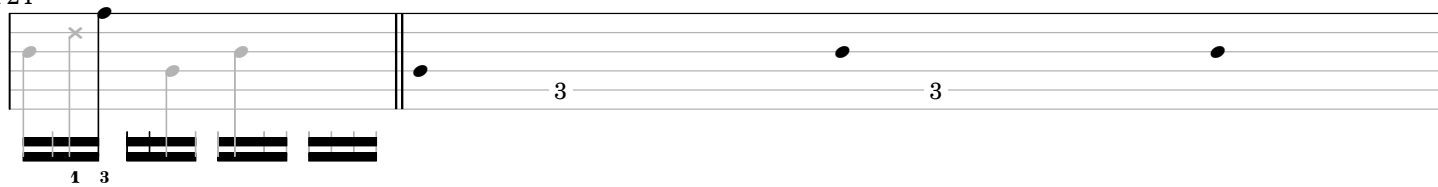


14'12"



14'20"

14'24"



[illegible]

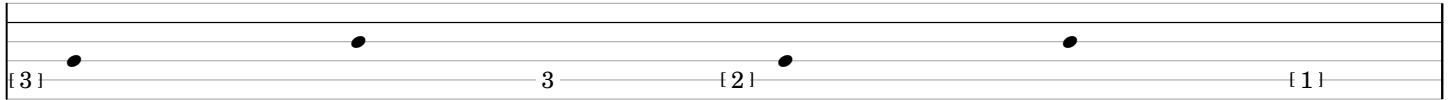
12

# *ostinato and interrupt*

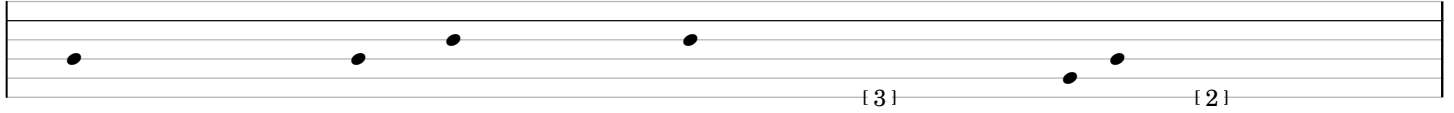
ensemble part 2

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

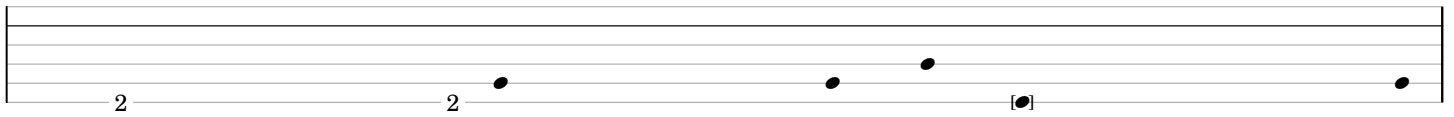
0'12"



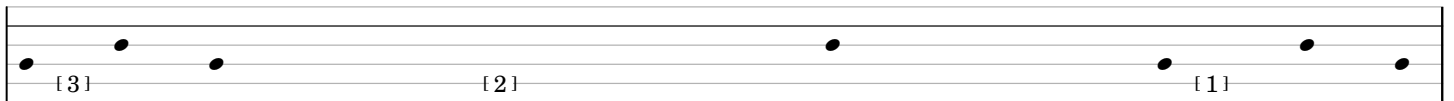
0'24"



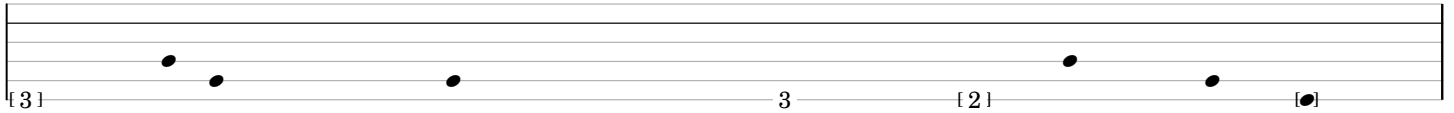
0'36"



0'48"



1'00"

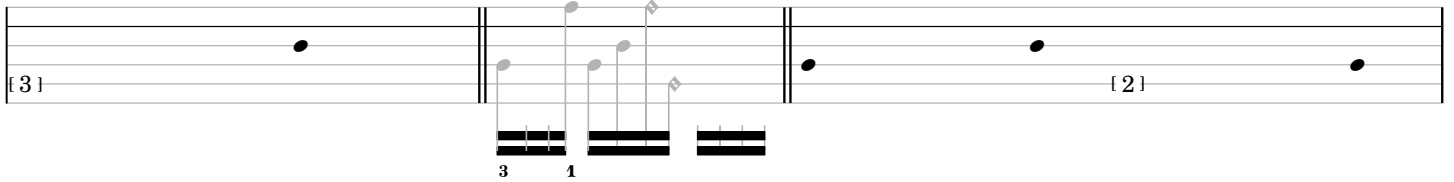


1'12"

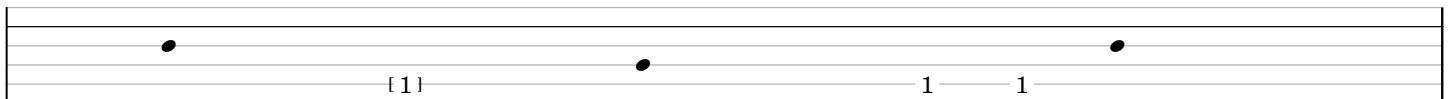


1'24"

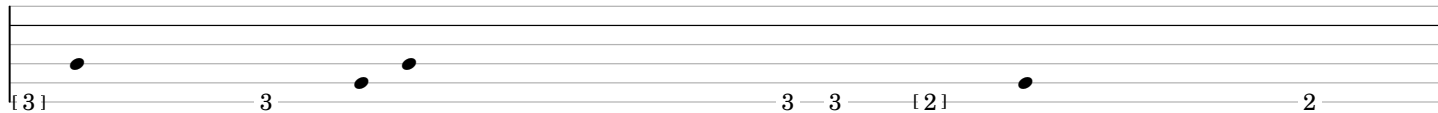
1'28"



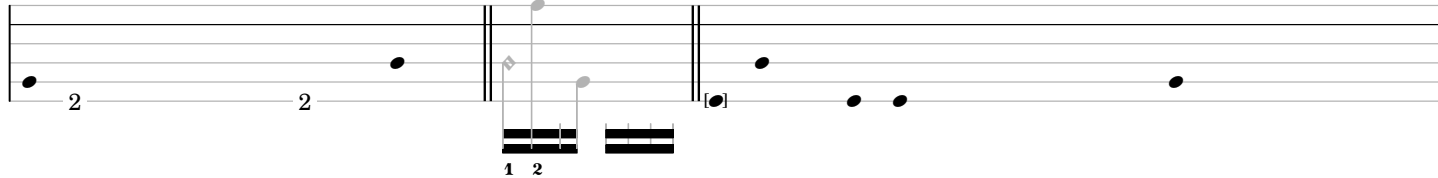
1'36"



1'48"



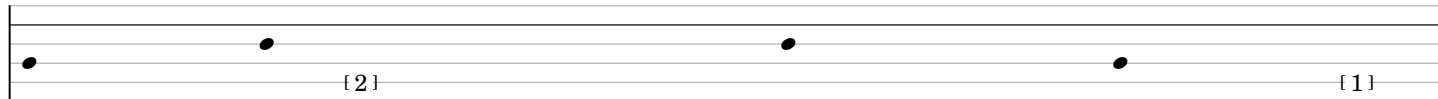
2'00"



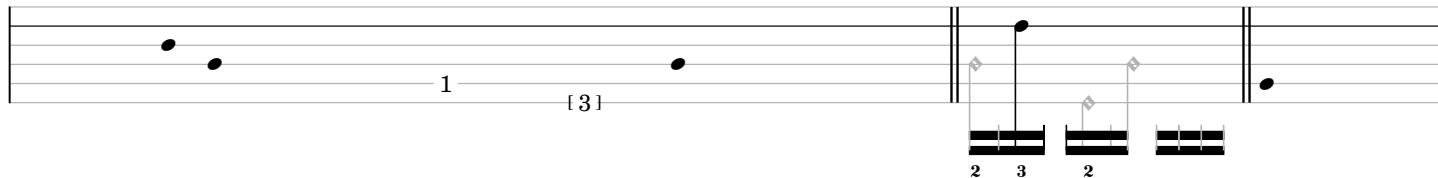
2'12"



2'24"



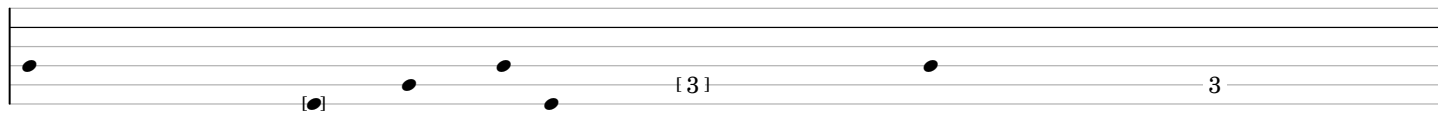
2'36"



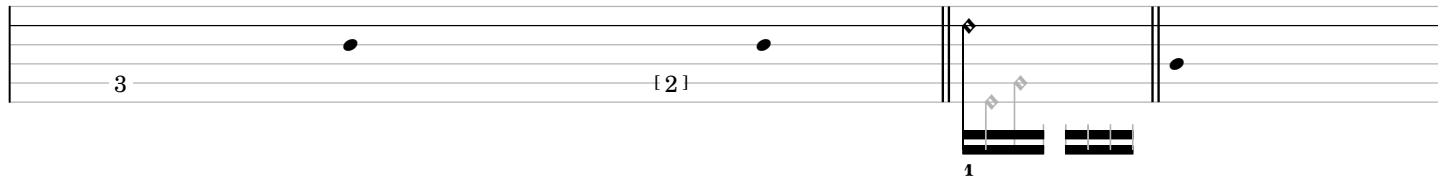
2'48"



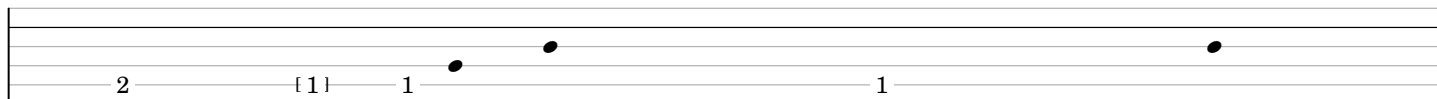
3'00"



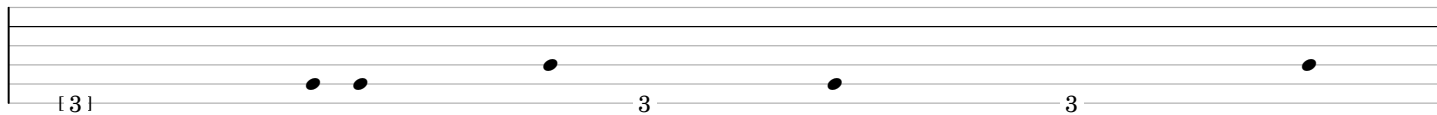
3'12"



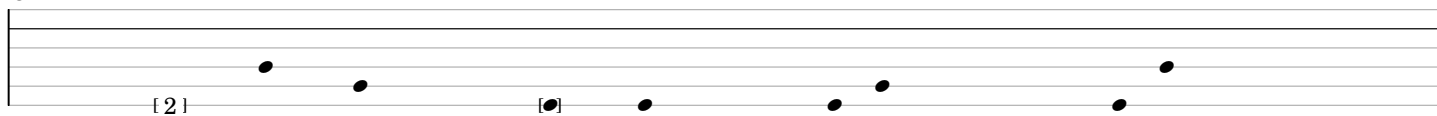
3'24"



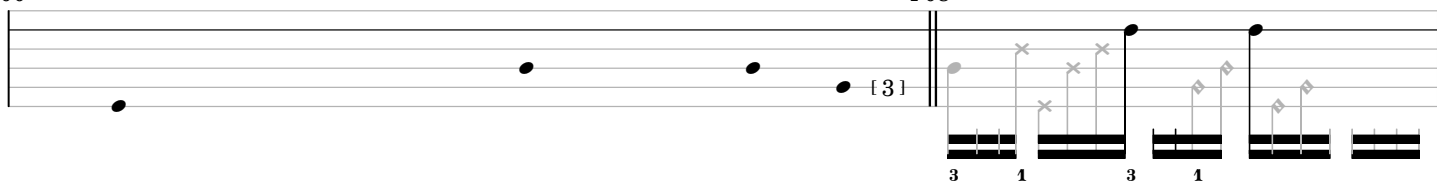
3'36"



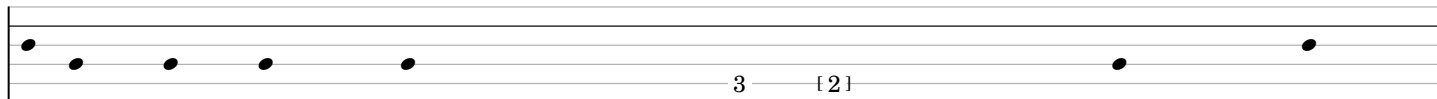
3'48"



4'00"



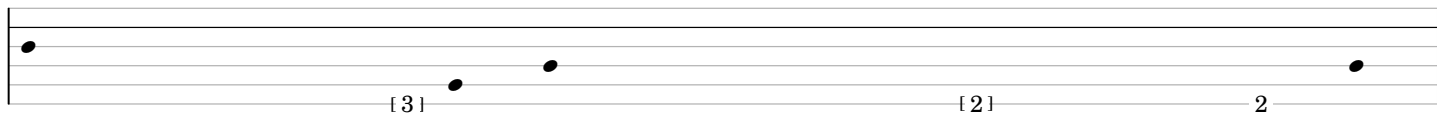
4'12"



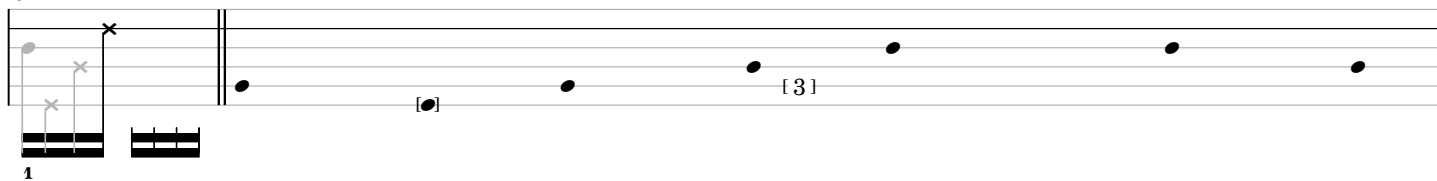
4'24"



4'36"



4'48"



5'00" 5'04"

5'12"

5'24"

5'36" 5'44"

5'48"

6'00"

6'12" 6'16"

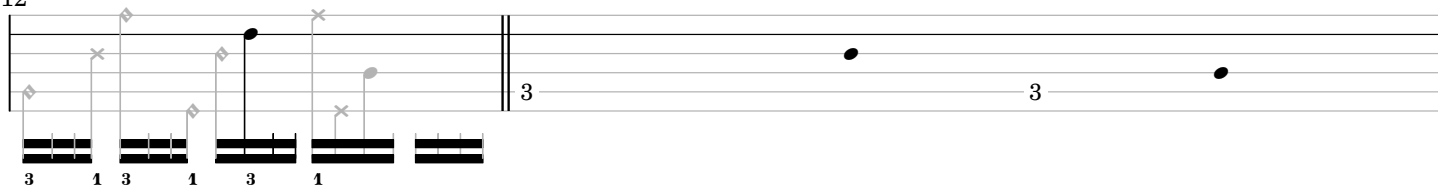
6'24"

[illegible]

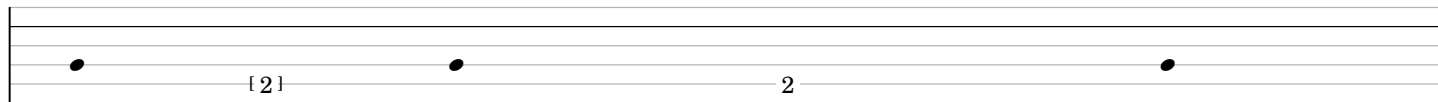
The first system of the musical score for 'The Little Boat' consists of two staves. The upper staff is a treble clef with a key signature of one flat (B-flat) and a 2/4 time signature. It contains a melody starting on a whole note G4, followed by a half note F4, and then a half note E4. The lower staff 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 first system of the musical score 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 on a whole note G4, followed by a half note F4, and then a quarter note E4. This is followed by a quarter rest, then a quarter note D4, and another quarter rest. The system concludes with a double bar line. Below the staff, the number '1' is written under the first G4 note, and the number '3' is written under the first D4 note, indicating fingerings.

8'12"

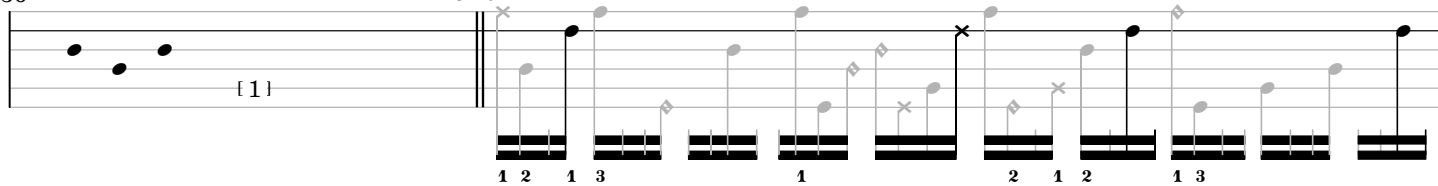


8'24"

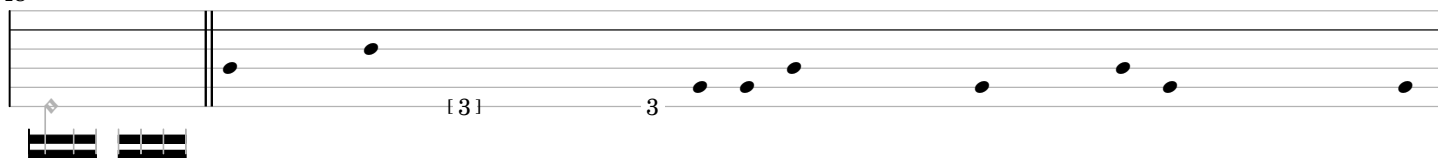


8'36"

8'40"

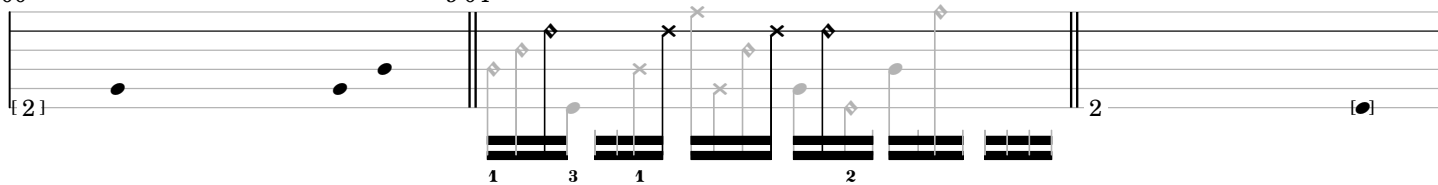


8'48"

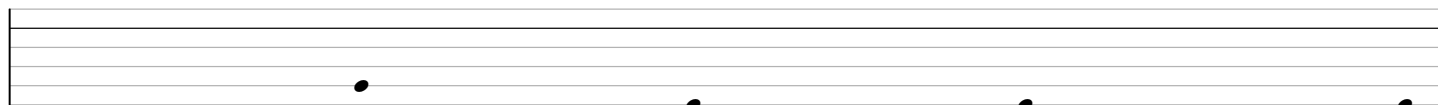


9'00"

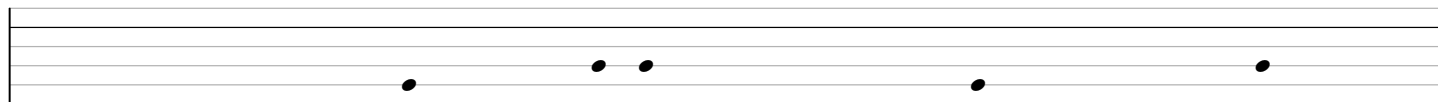
9'04"



9'12"

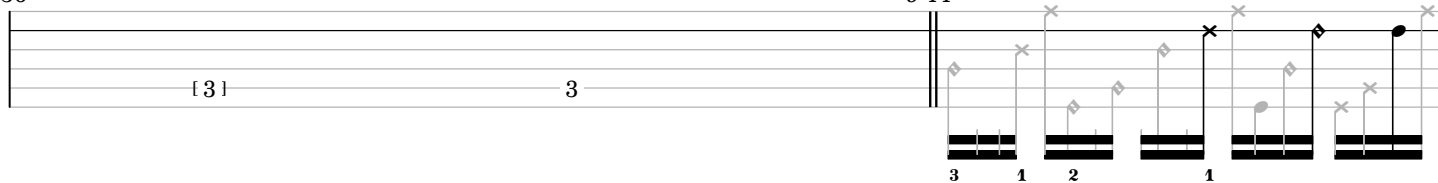


9'24"



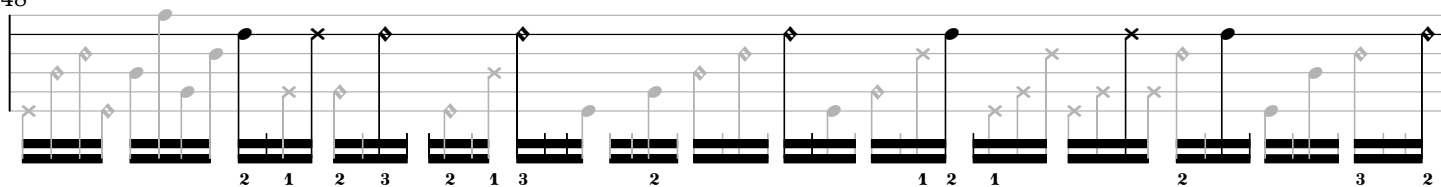
9'36"

9'44"

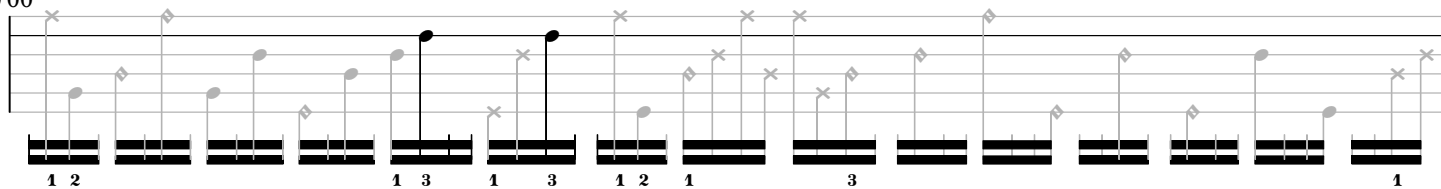




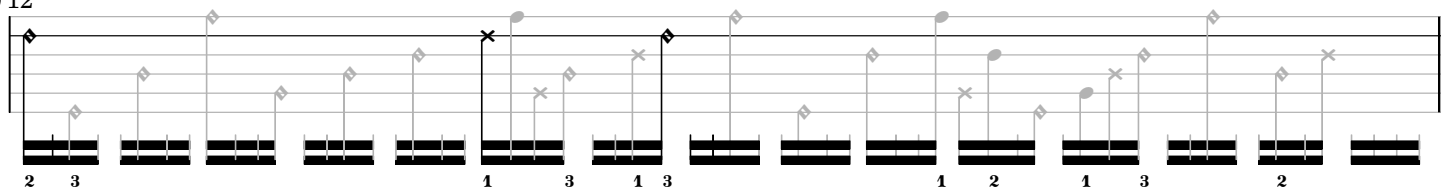
9'48"



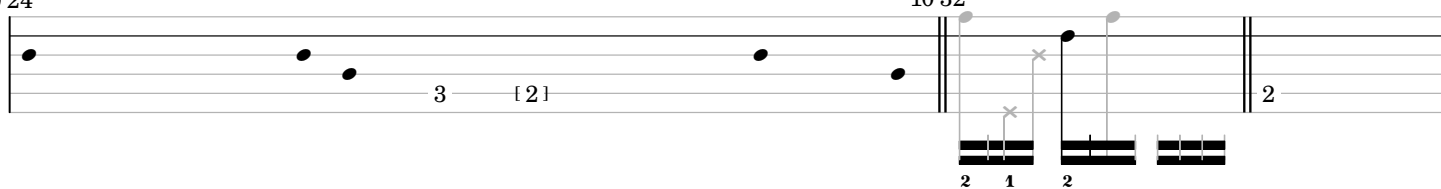
10'00"



10'12"

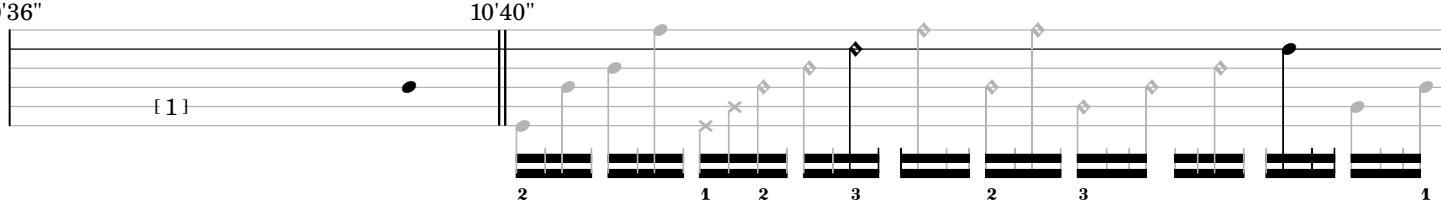


10'24"



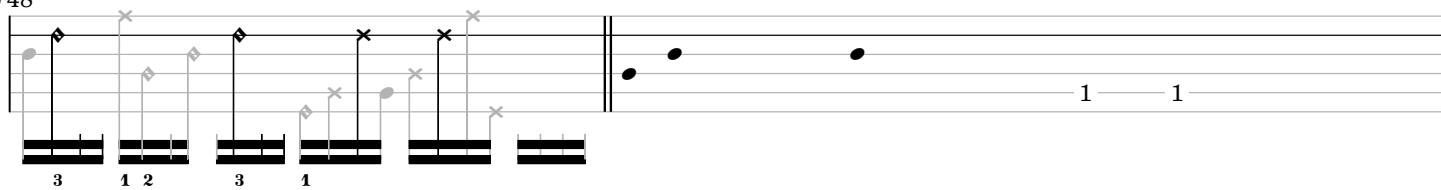
10'32"

10'36"

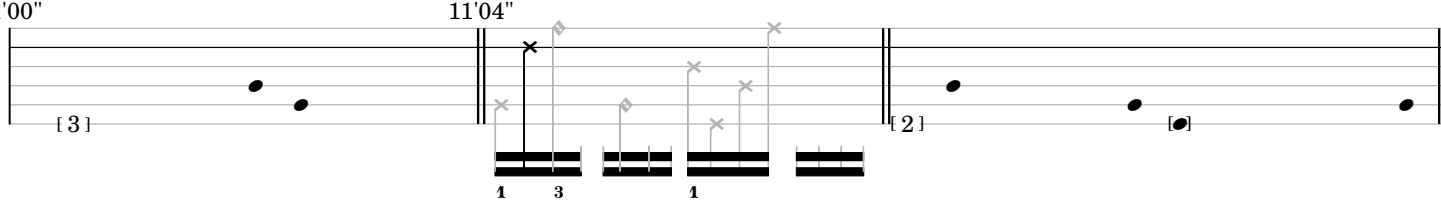


10'40"

10'48"

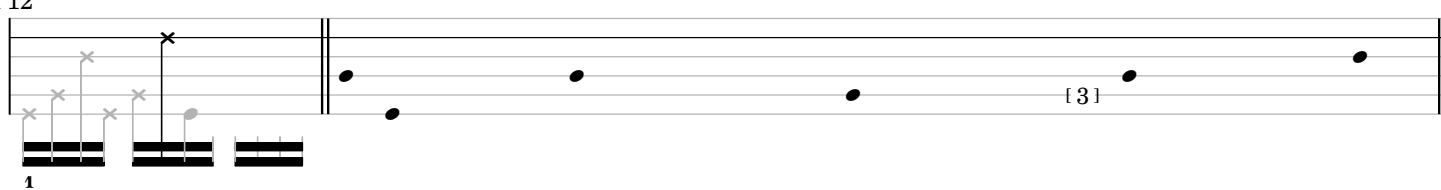


11'00"



11'04"

11'12"



11'24"

Musical notation for 11'24". The staff shows a sequence of notes with various articulations (accents, slurs, and 'x' marks). The bottom staff contains fingerings: 2, 1 3, 2 3, 2 1 3, 1 2, 1. The notation ends with a double bar line and a repeat sign, followed by a measure with a [2] bracket and a measure with a [1] bracket.

11'36"

Musical notation for 11'36". The staff shows a sequence of notes with various articulations. The bottom staff contains fingerings: 1 2, 1. The notation ends with a double bar line and a repeat sign, followed by a measure with a [3] bracket. The time signature 11'44" is indicated above the staff.

11'44"

11'48"

Musical notation for 11'48". The staff shows a sequence of notes with various articulations. The bottom staff contains fingerings: 3 [2]. The notation ends with a double bar line and a repeat sign, followed by a measure with a [2] bracket and a measure with a [1] bracket.

12'00"

Musical notation for 12'00". The staff shows a sequence of notes with various articulations. The bottom staff contains fingerings: 1 2 1, 2, 1 2, 1, 2, 1 2, 1, 2, 1 2, 1, 2, 1. The notation ends with a double bar line and a repeat sign, followed by a measure with a [2] bracket and a measure with a [1] bracket.

12'12"

Musical notation for 12'12". The staff shows a sequence of notes with various articulations. The bottom staff contains fingerings: 3. The notation ends with a double bar line and a repeat sign, followed by a measure with a [3] bracket.

12'24"

Musical notation for 12'24". The staff shows a sequence of notes with various articulations. The bottom staff contains fingerings: 2, 1. The notation ends with a double bar line and a repeat sign, followed by a measure with a [2] bracket.

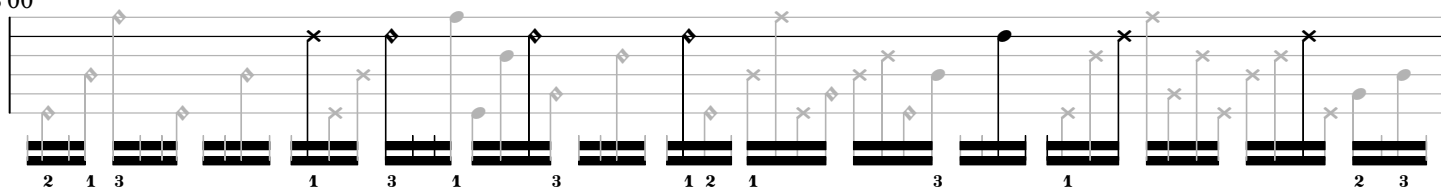
12'36"

Musical notation for 12'36". The staff shows a sequence of notes with various articulations. The bottom staff contains fingerings: 3, 1. The notation ends with a double bar line and a repeat sign, followed by a measure with a [2] bracket and a measure with a [1] bracket.

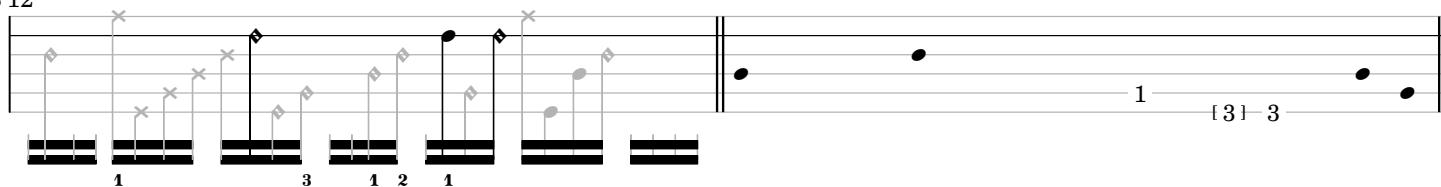
12'48"

Musical notation for 12'48". The staff shows a sequence of notes with various articulations. The bottom staff contains fingerings: 2, 3, 1 2 1, 3, 1 3, 1 3, 1 3. The notation ends with a double bar line and a repeat sign, followed by a measure with a [2] bracket and a measure with a [1] bracket.

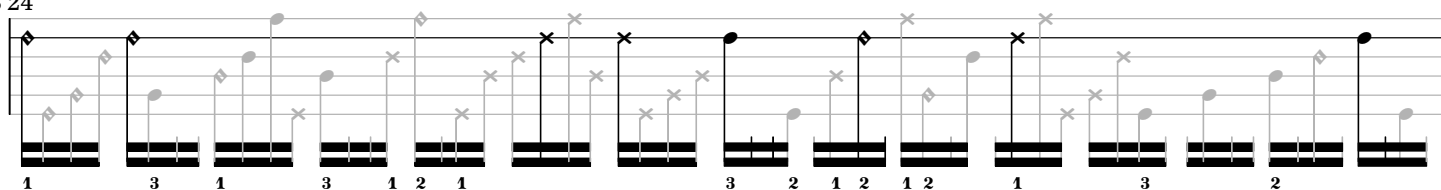
13'00"



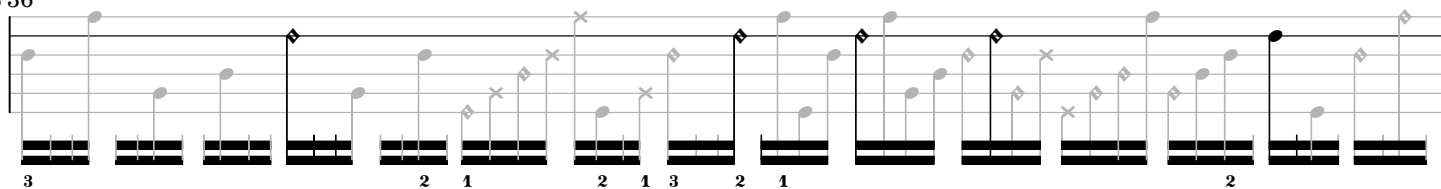
13'12"



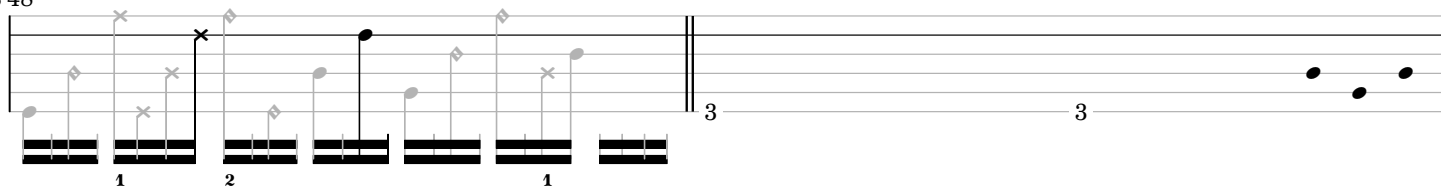
13'24"



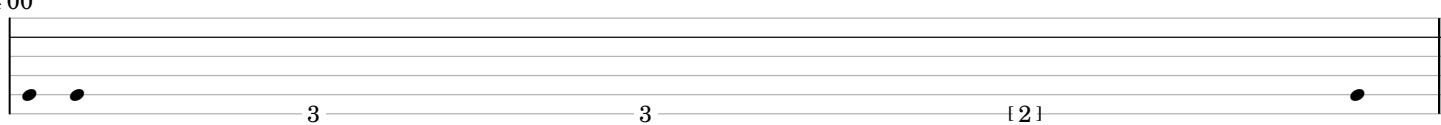
13'36"



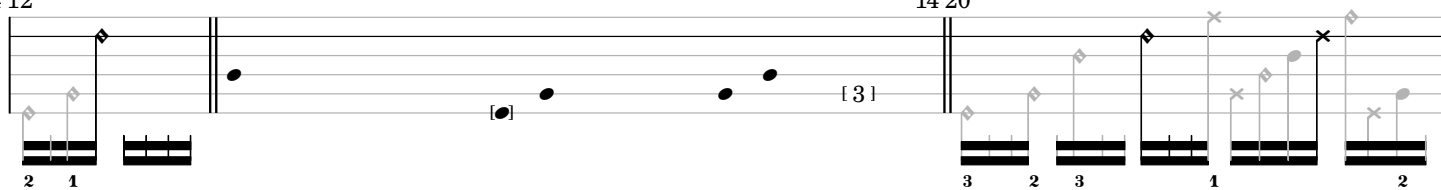
13'48"



14'00"

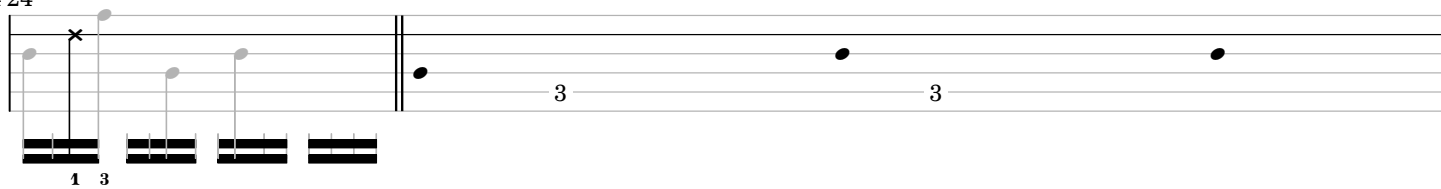


14'12"



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"

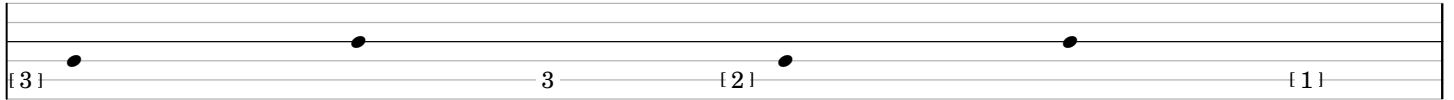
[2] 2 [2]

# *ostinato and interrupt*

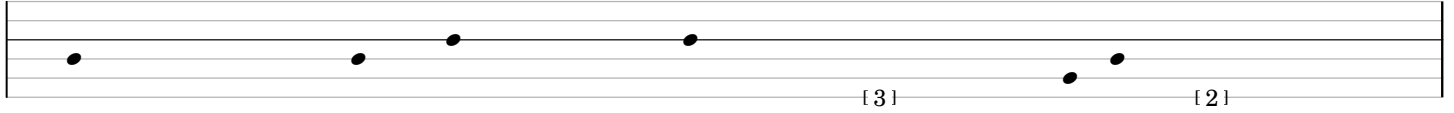
ensemble part 3

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

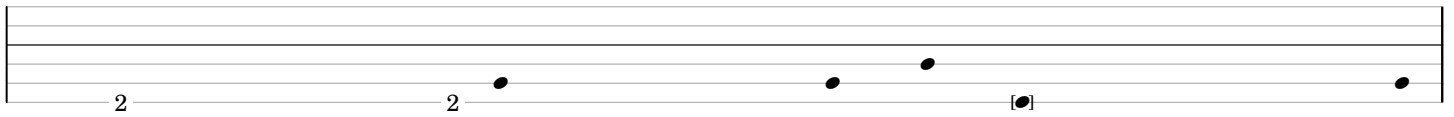
0'12"



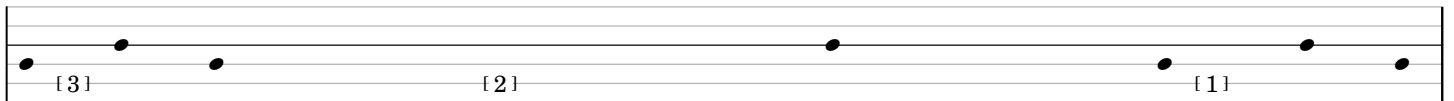
0'24"



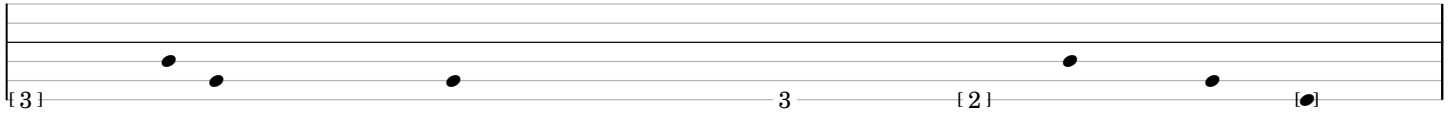
0'36"



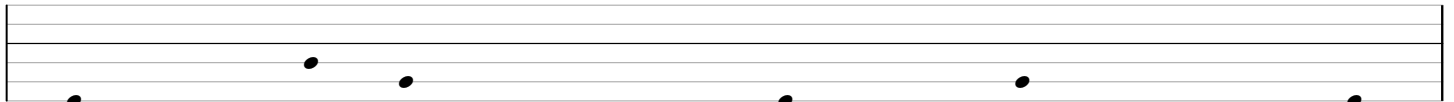
0'48"



1'00"

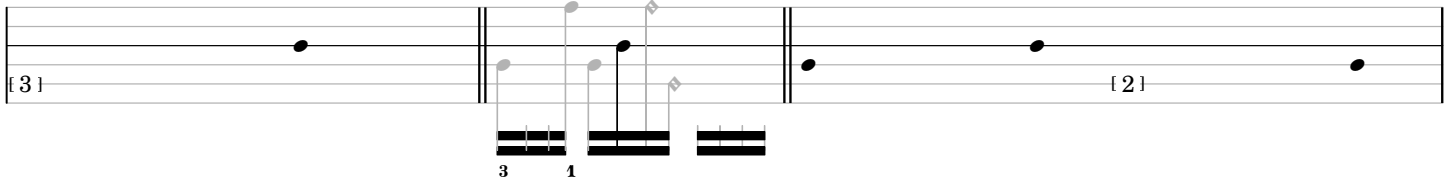


1'12"

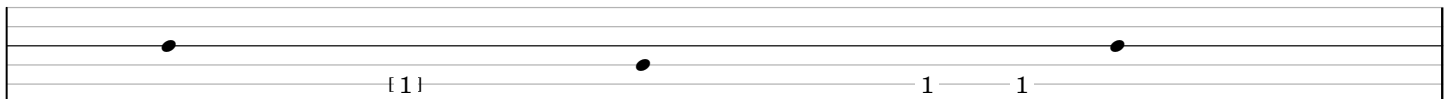


1'24"

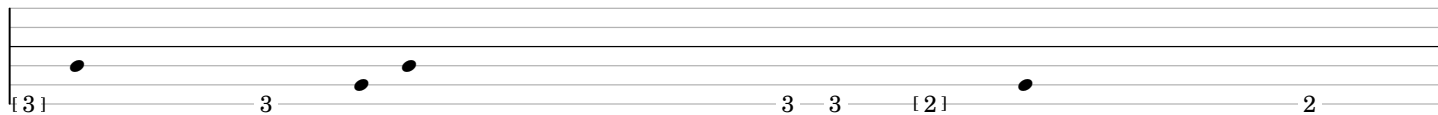
1'28"



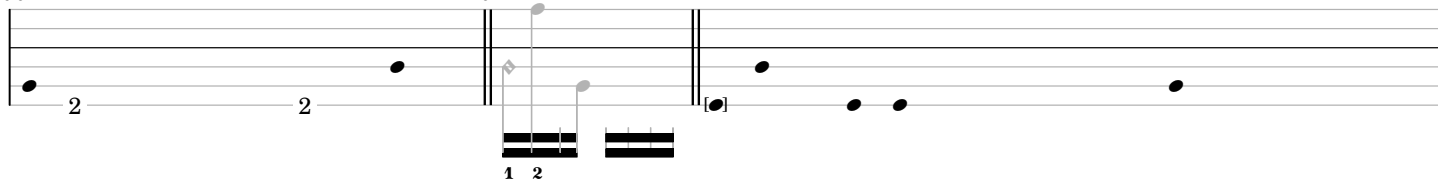
1'36"



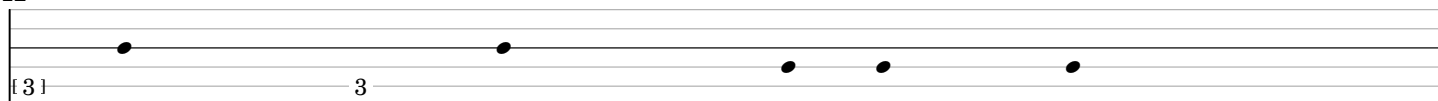
1'48"



2'00"



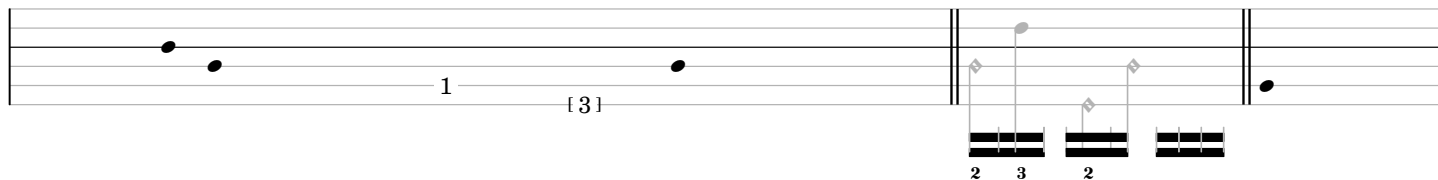
2'12"



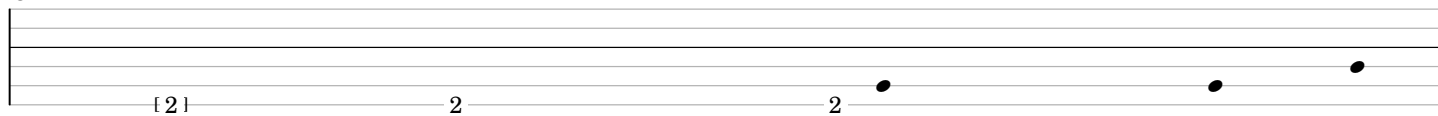
2'24"



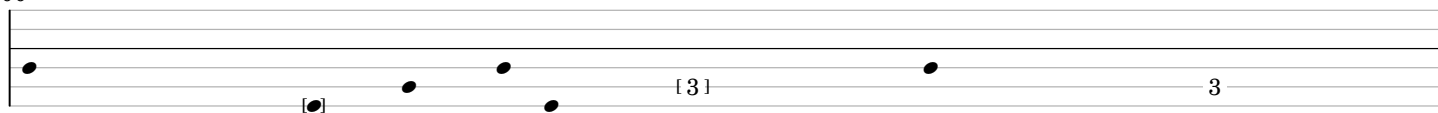
2'36"



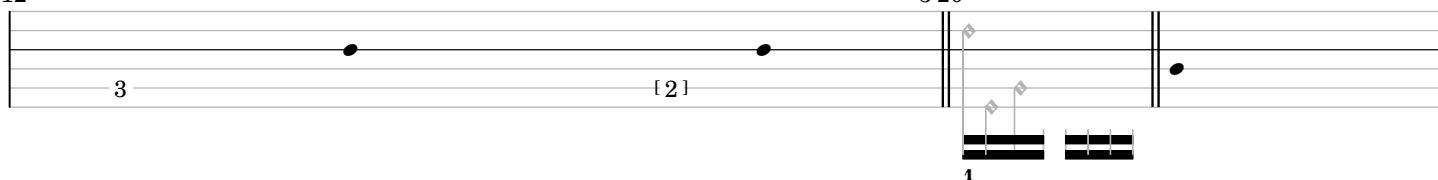
2'48"



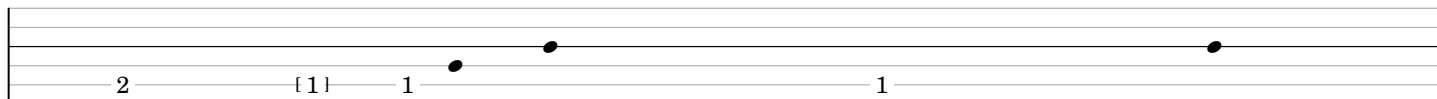
3'00"



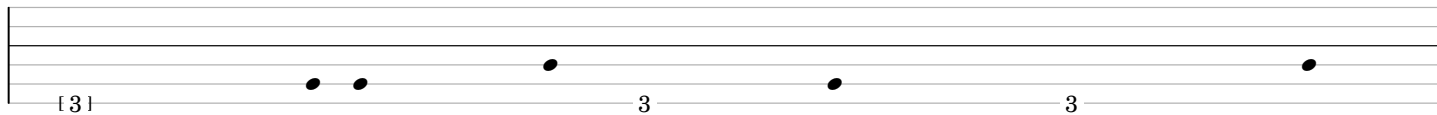
3'12"



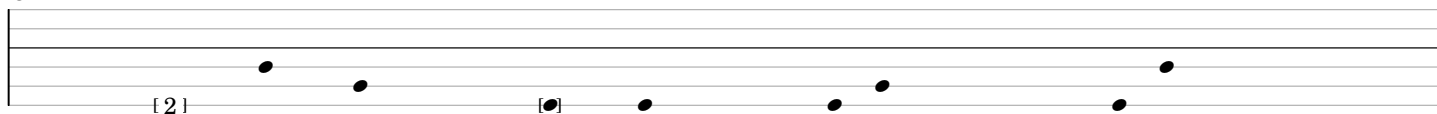
3'24"



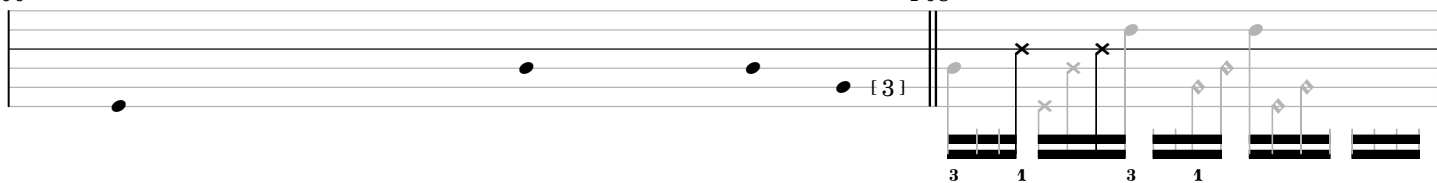
3'36"



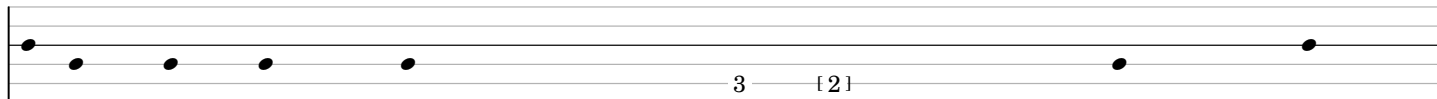
3'48"



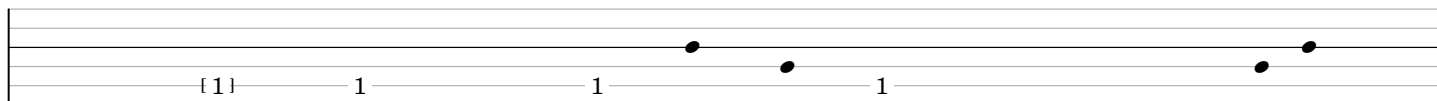
4'00"



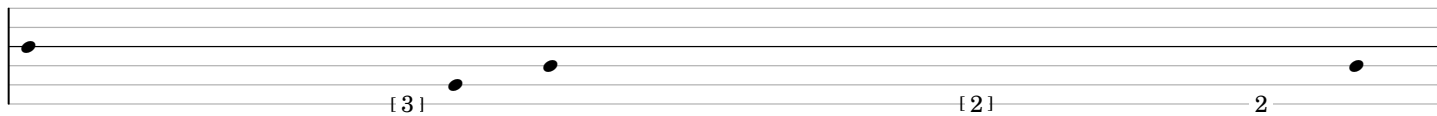
4'12"



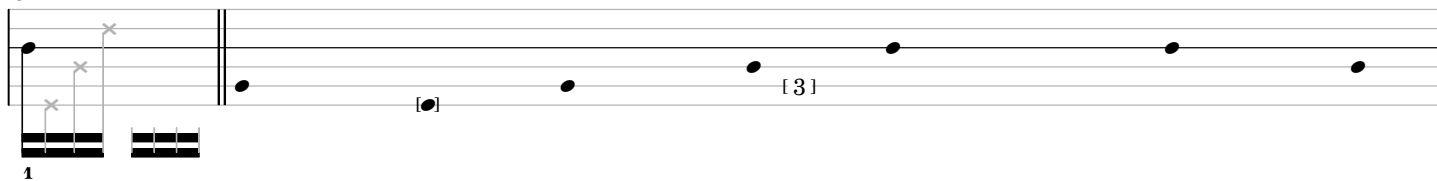
4'24"



4'36"



4'48"



5'00" 5'04"

[2] 2 1

5'12"

2 [1]

5'24"

1 [3] 3 3

5'36" 5'44"

3 [2] 1 2 1 3 1 3

5'48"

2 1 2 [2]

6'00"

[3] 3 3 [2]

6'12" 6'16"

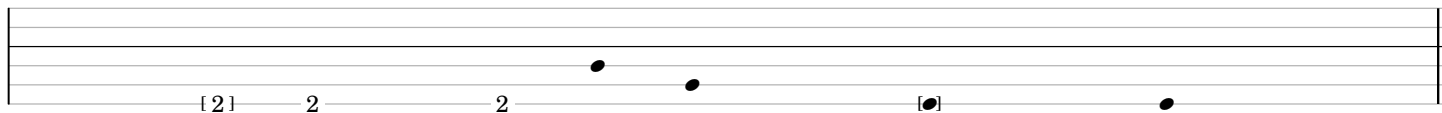
[1] 1 2 1 2 1 1

6'24"

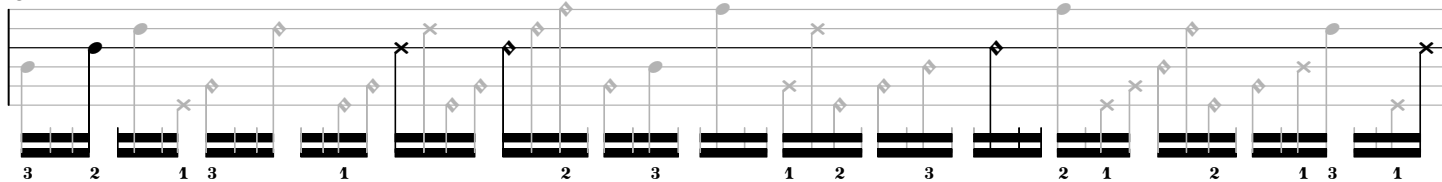
[3] 3



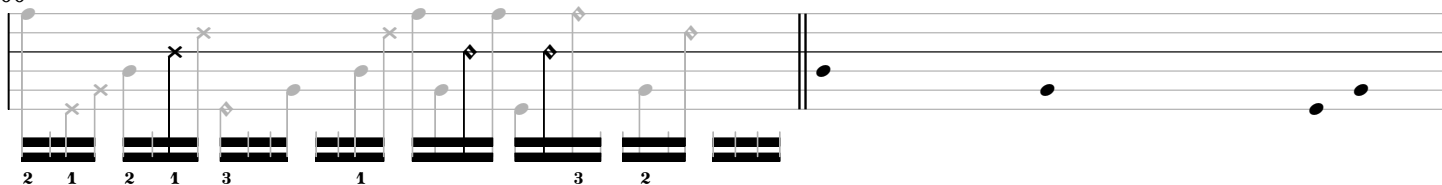
6'36"



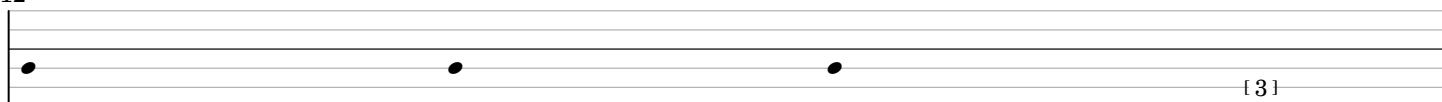
6'48"



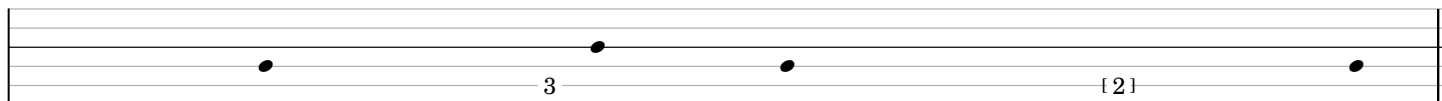
7'00"



7'12"



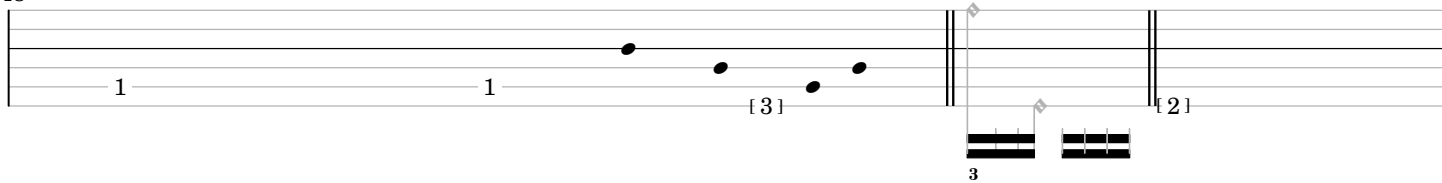
7'24"



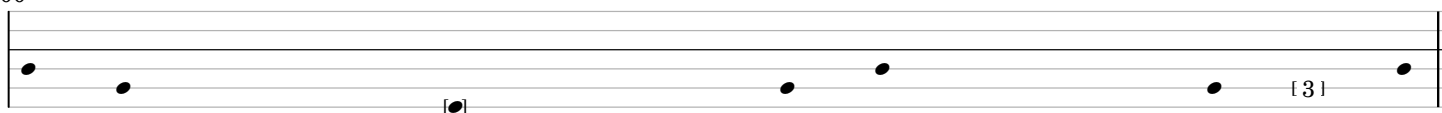
7'36"



7'48"



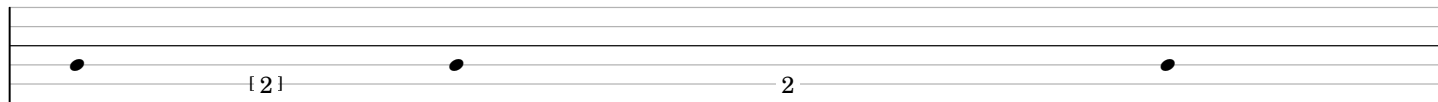
8'00"



8'12"

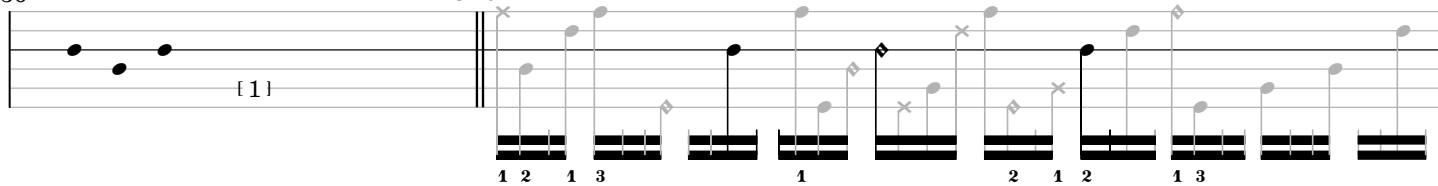


8'24"

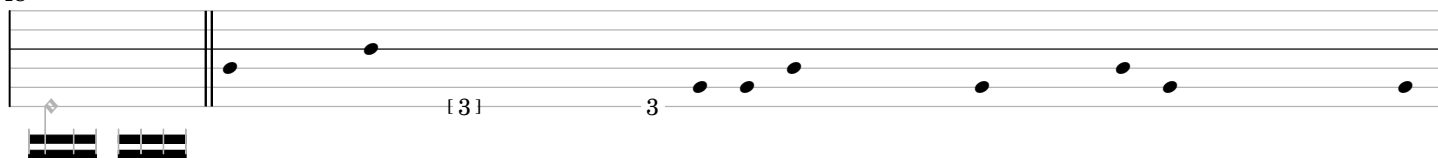


8'36"

8'40"

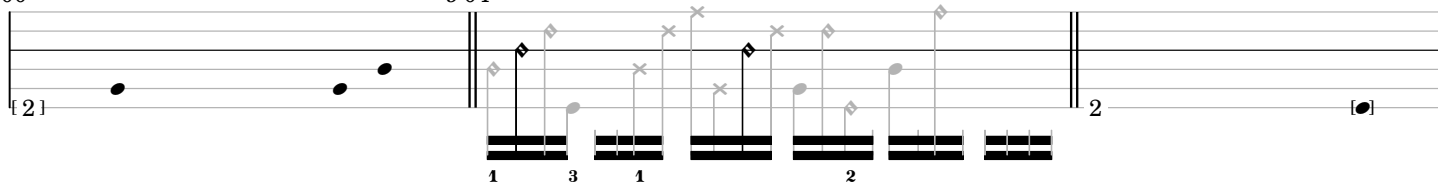


8'48"

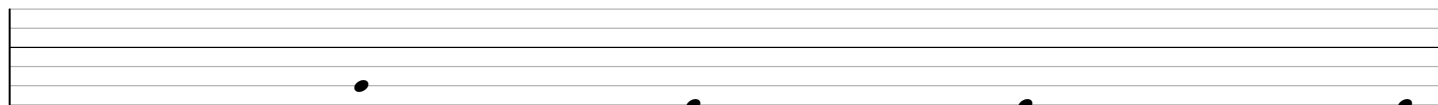


9'00"

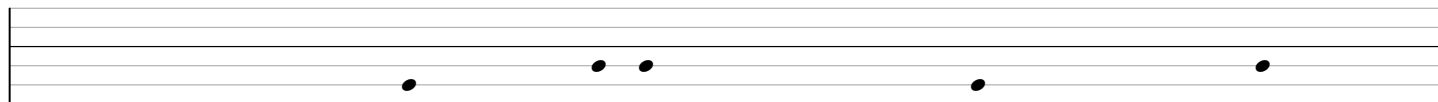
9'04"



9'12"



9'24"

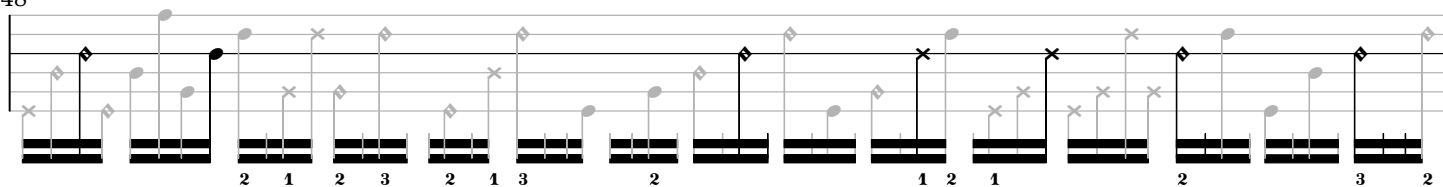


9'36"

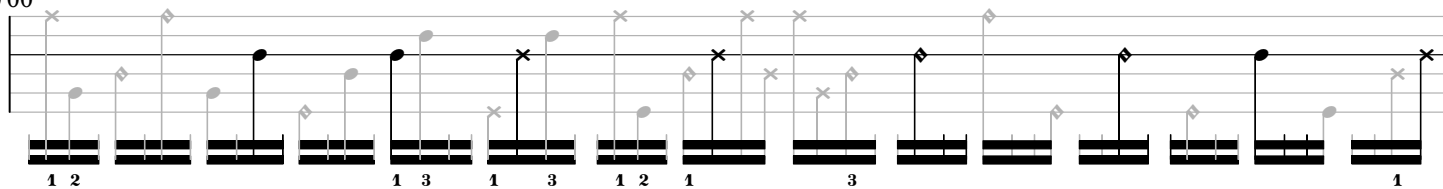
9'44"



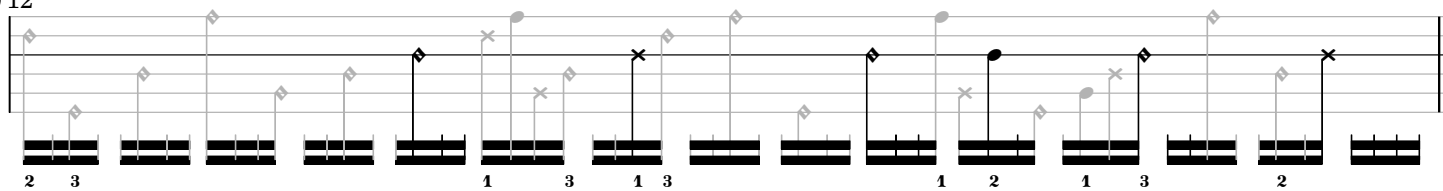
9'48"



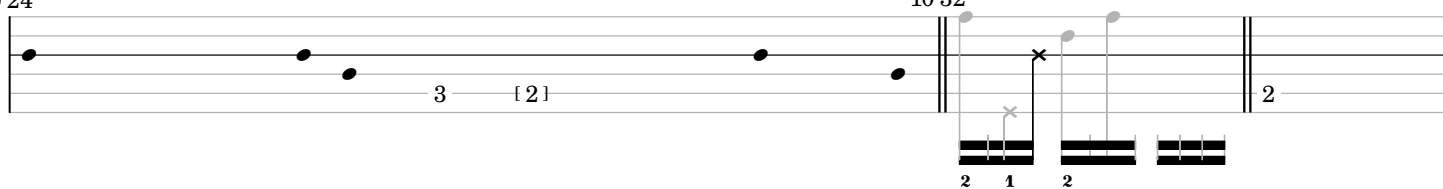
10'00"



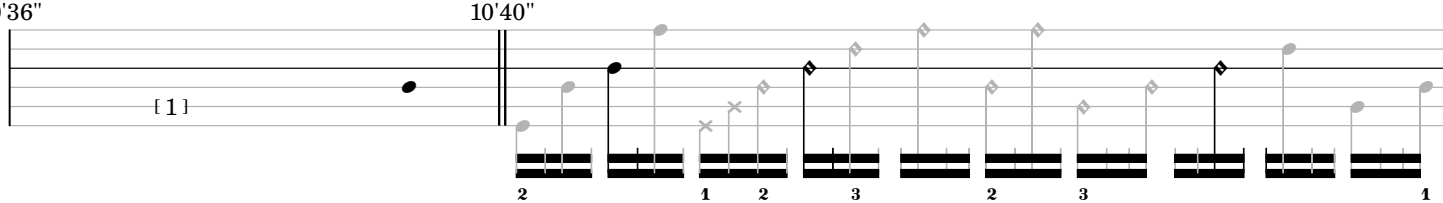
10'12"



10'24"

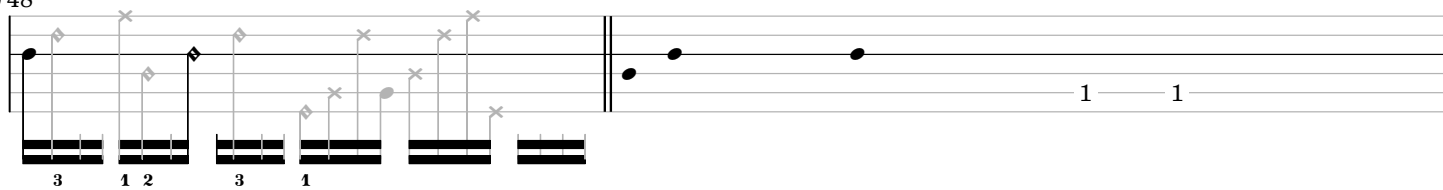


10'36"

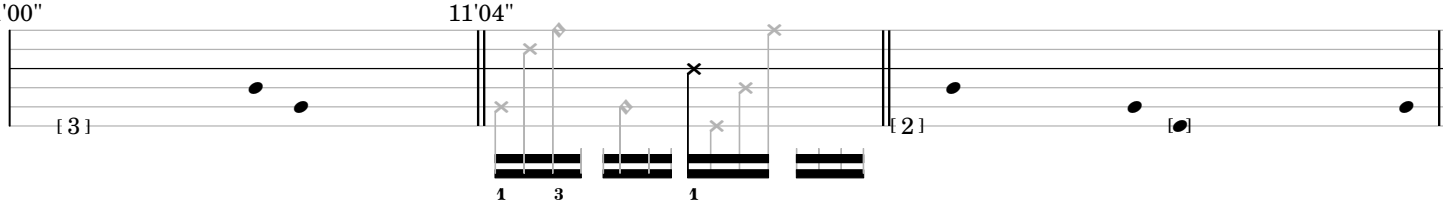


10'40"

10'48"

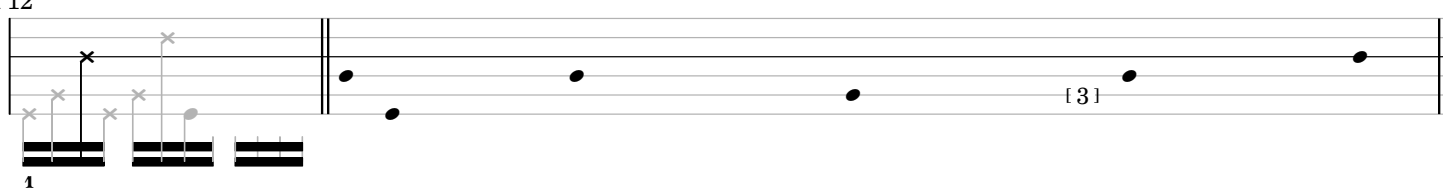


11'00"



11'04"

11'12"



11'24"

2 1 3 2 3 2 1 3 1 2 1

11'36"

1 2 1

11'44"

1 3 1

11'48"

3 [2]

12'00"

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

12'12"

[3]

12'24"

2 1

12'36"

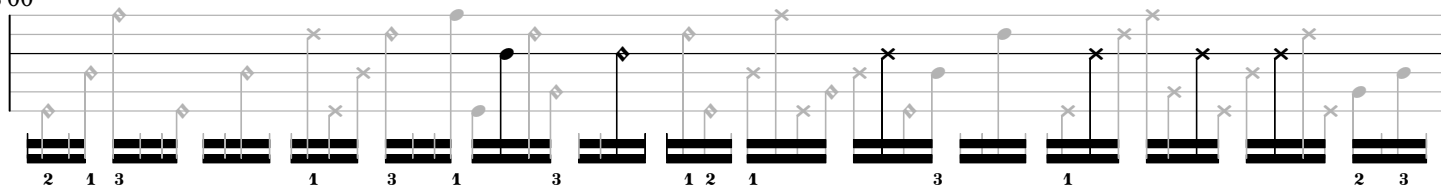
3 1

2 [1] 1

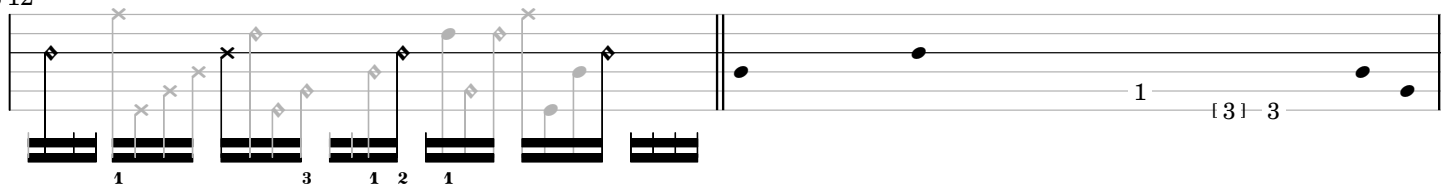
12'48"

2 3 1 2 1 3 1 3 1 3

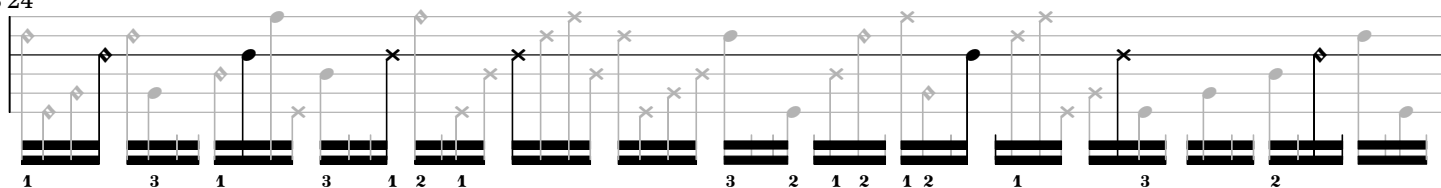
13'00"



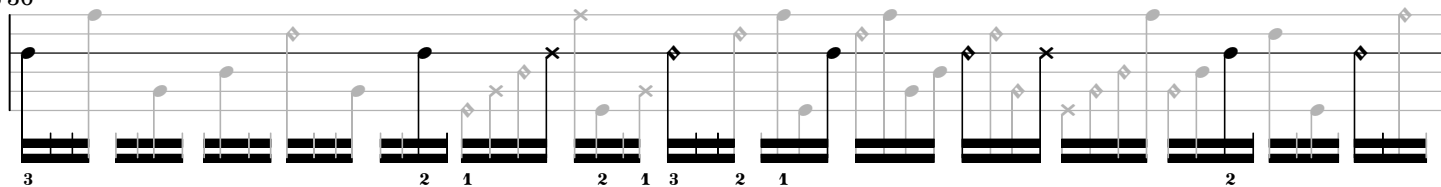
13'12"



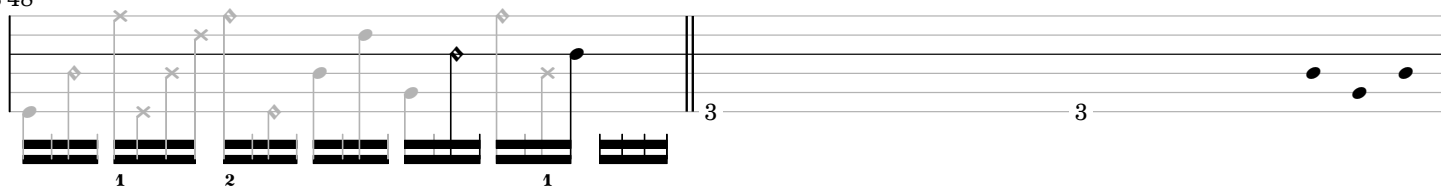
13'24"



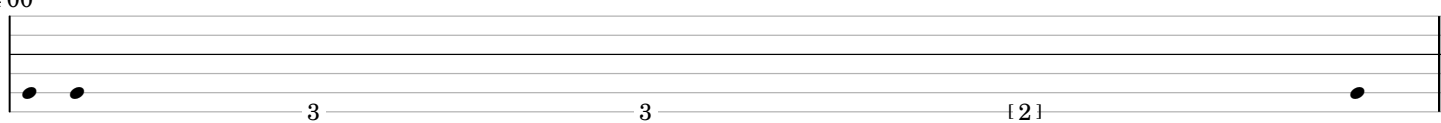
13'36"



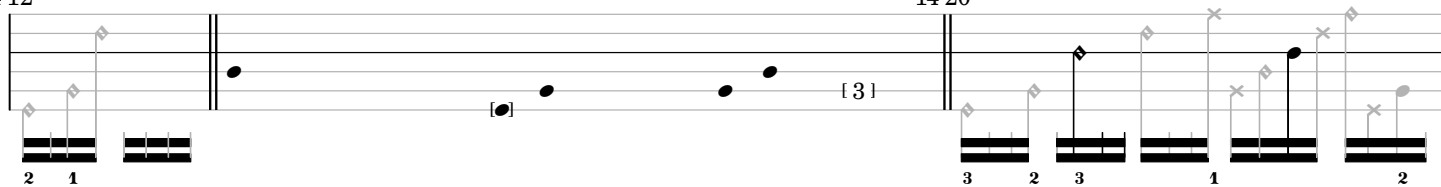
13'48"



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 2 2 [3]

15'36"

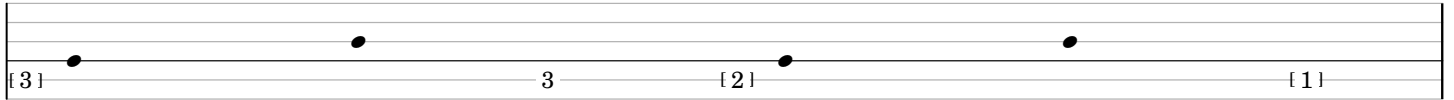
[2] 2 [6]

# *ostinato and interrupt*

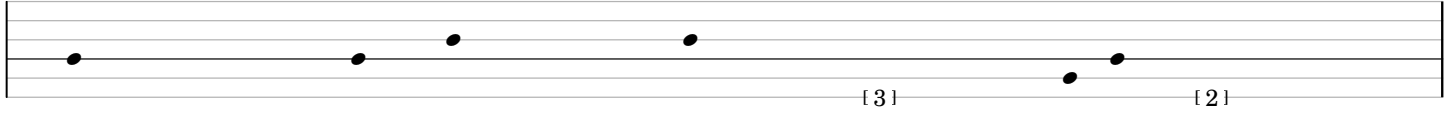
ensemble part 4

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

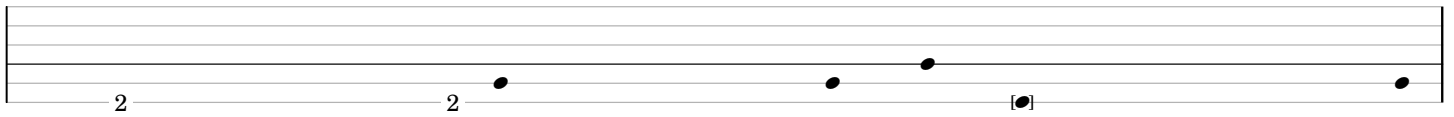
0'12"



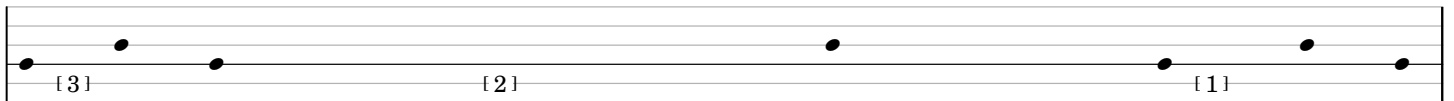
0'24"



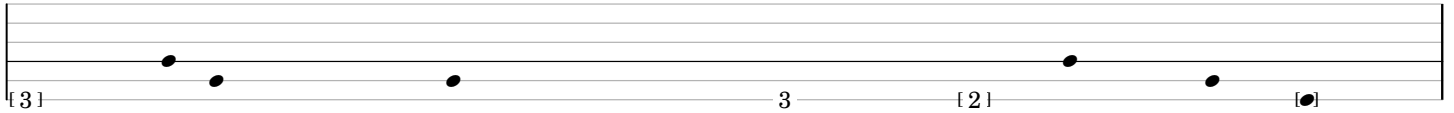
0'36"



0'48"



1'00"

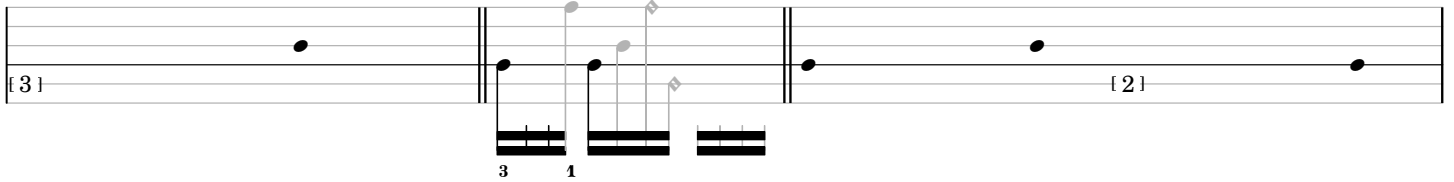


1'12"

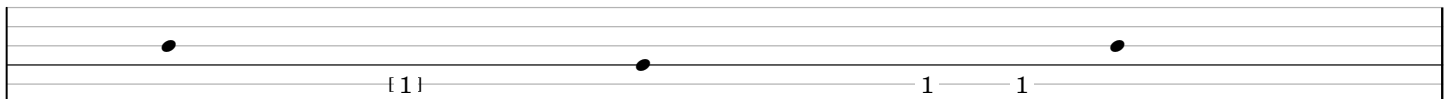


1'24"

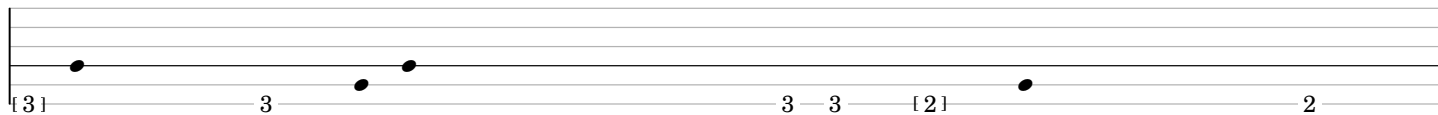
1'28"



1'36"

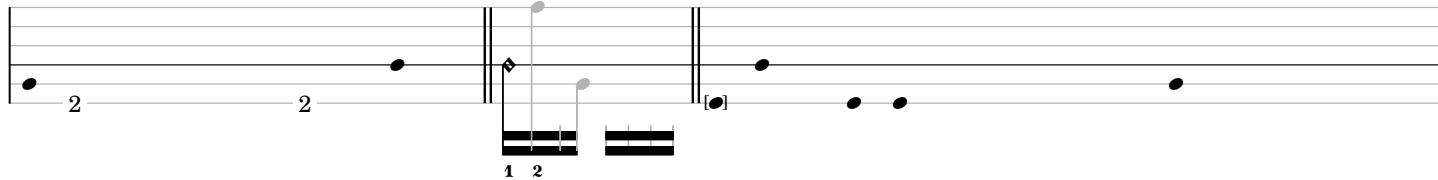


1'48"

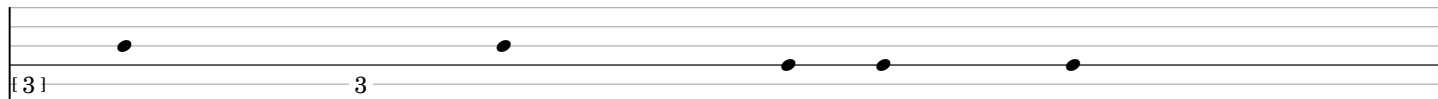


2'00"

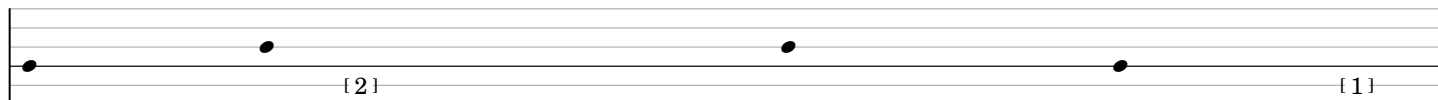
2'04"



2'12"

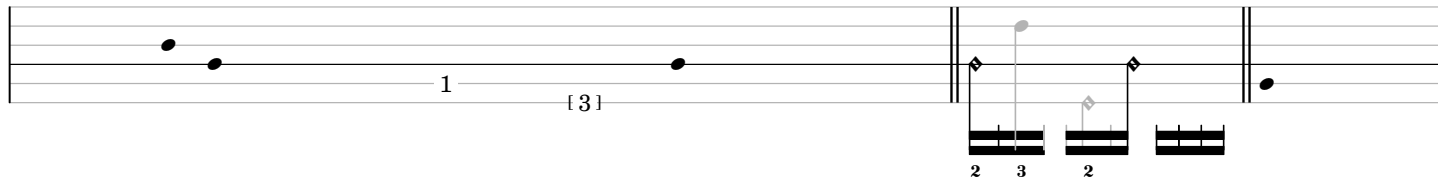


2'24"

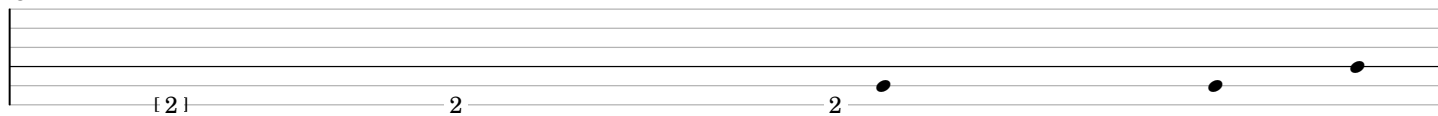


2'36"

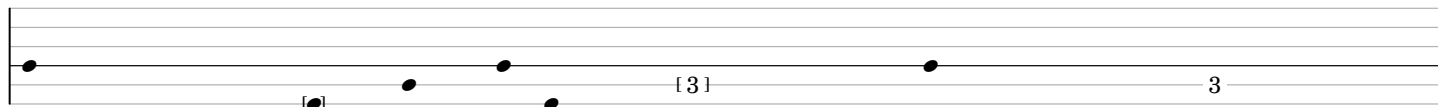
2'44"



2'48"

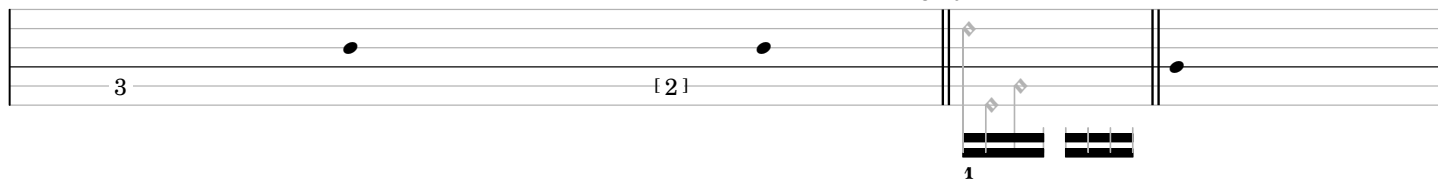


3'00"



3'12"

3'20"







5'00" 5'04"

5'12"

5'24"

5'36" 5'44"

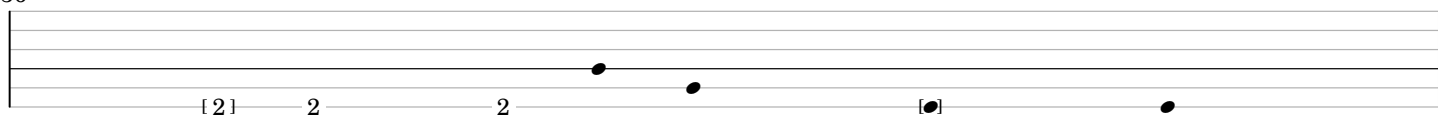
5'48"

6'00"

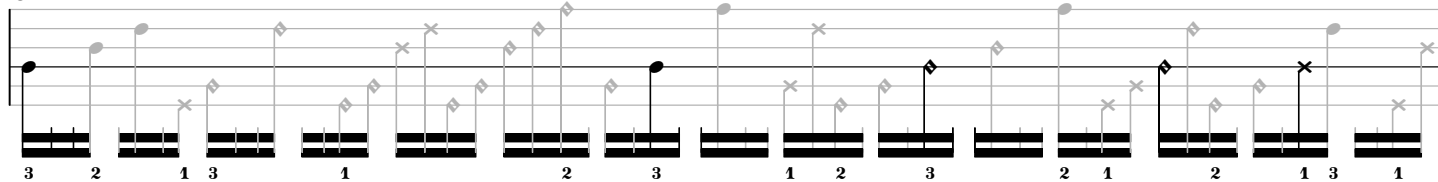
6'12" 6'16"

6'24"

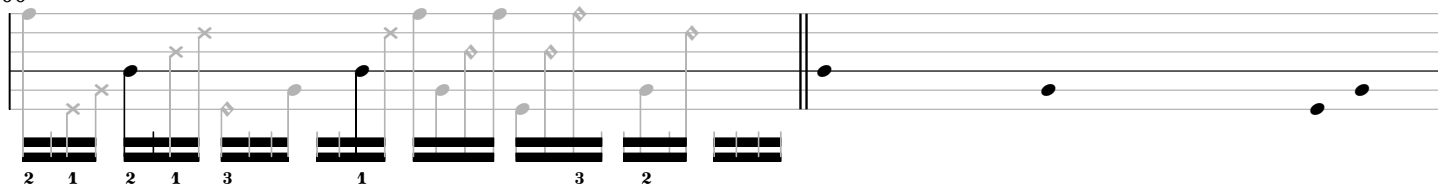
6'36"



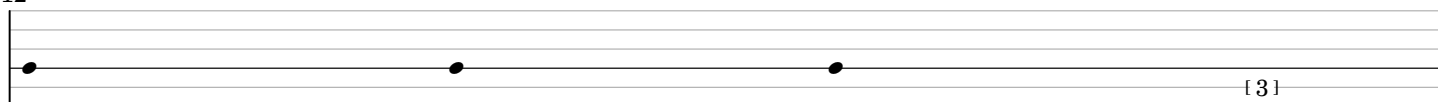
6'48"



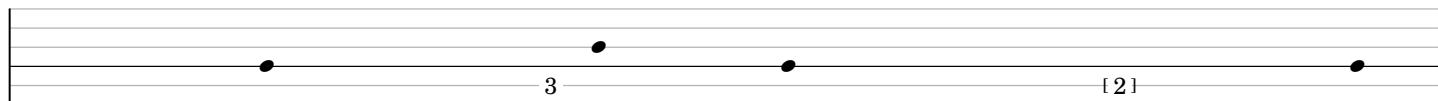
7'00"



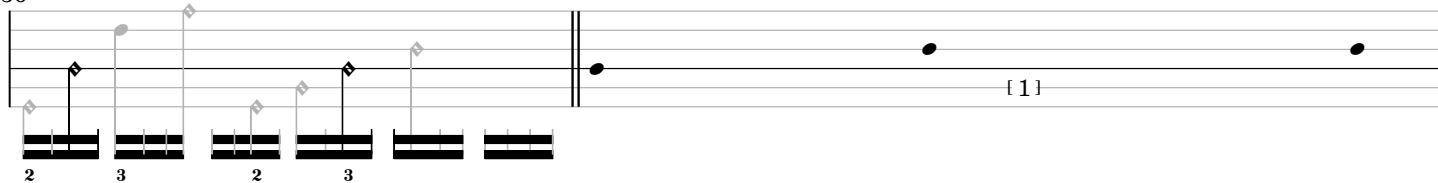
7'12"



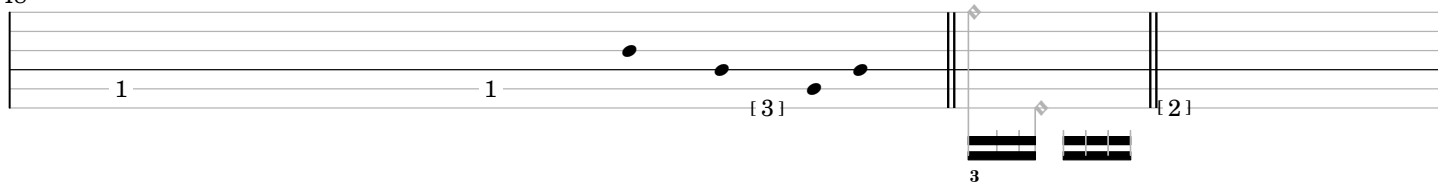
7'24"



7'36"

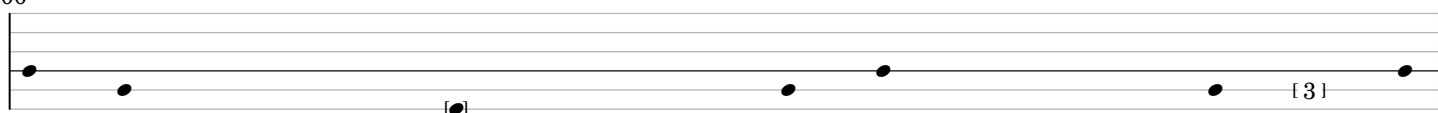


7'48"

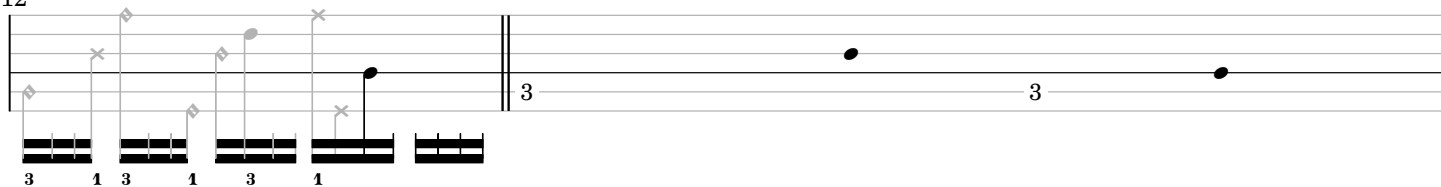


7'56"

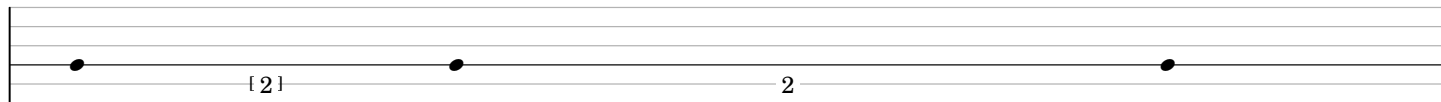
8'00"



8'12"

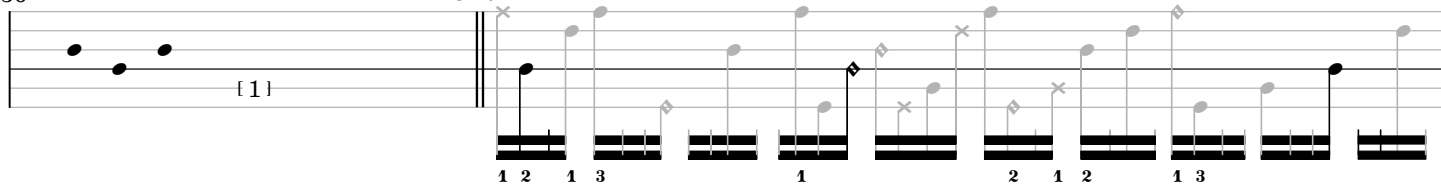


8'24"

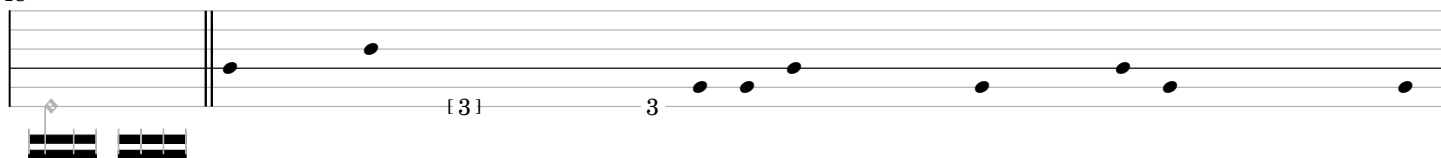


8'36"

8'40"

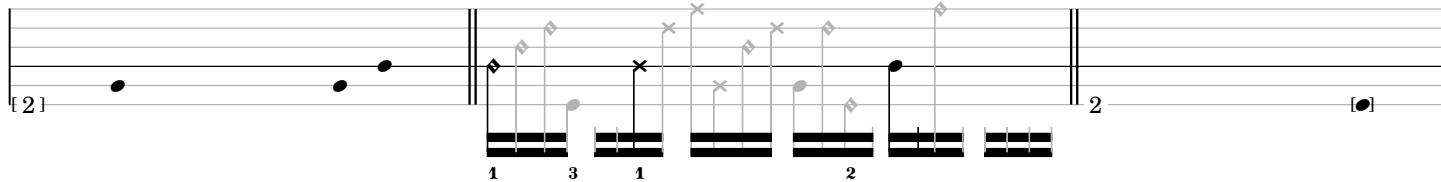


8'48"

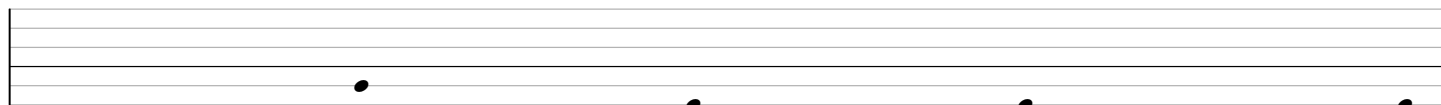


9'00"

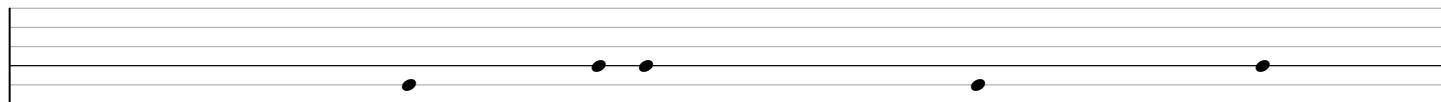
9'04"



9'12"



9'24"

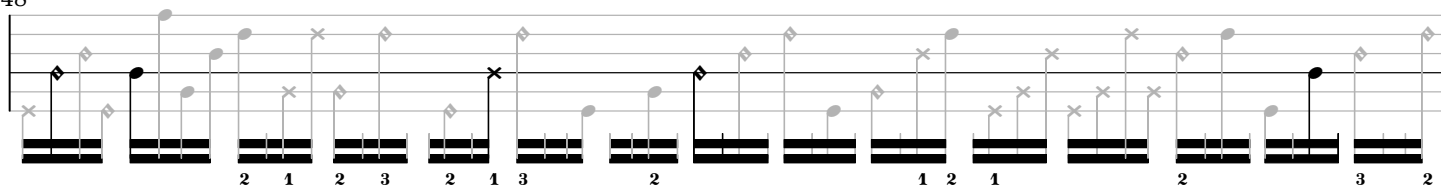


9'36"

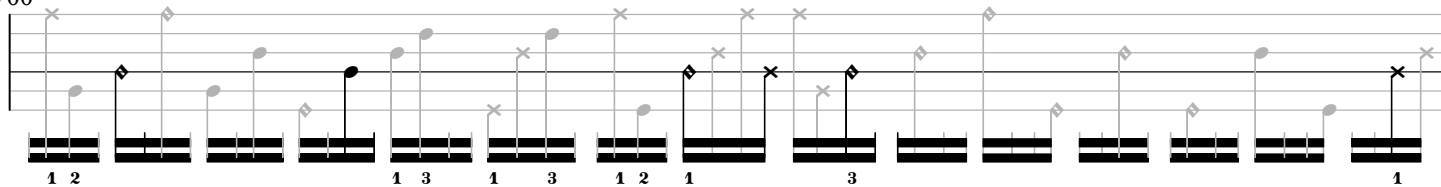
9'44"



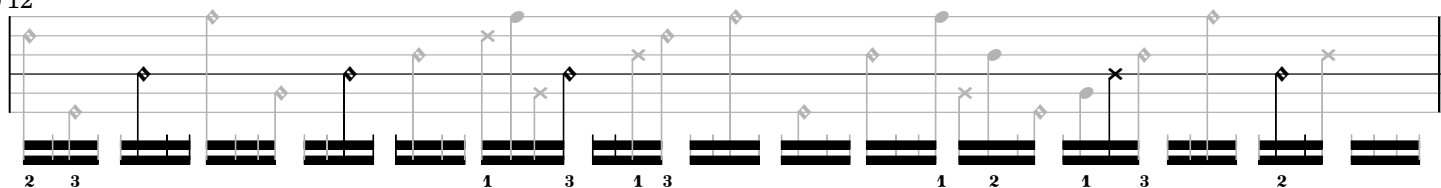
9'48"



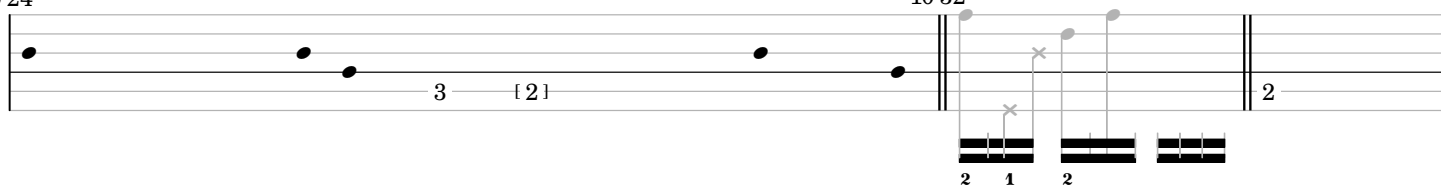
10'00"



10'12"



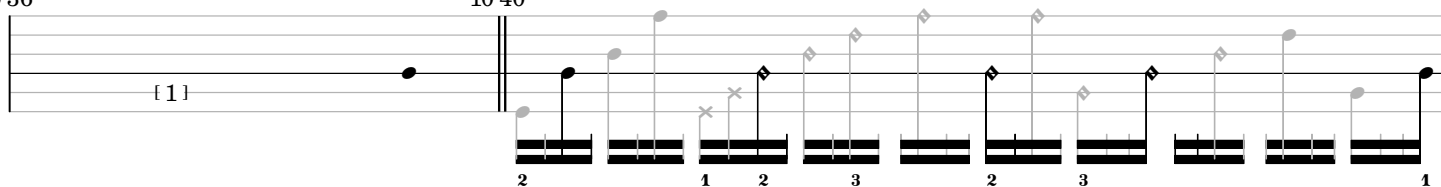
10'24"



10'32"

10'36"

10'40"

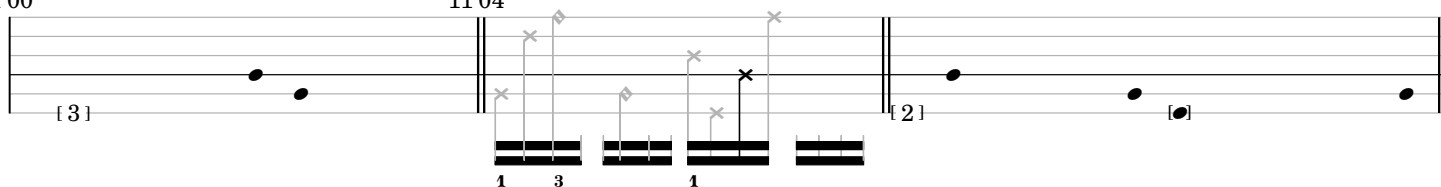


10'48"

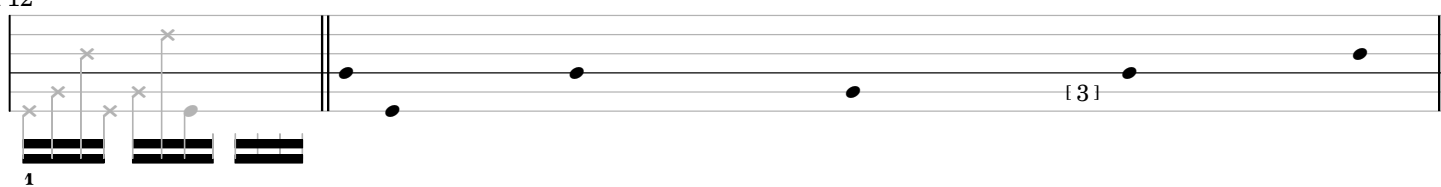


11'00"

11'04"



11'12"



11'24"

2 1 3 2 3 2 1 3 1 2 1

11'36"

1 2 1

11'44"

1 3 1

11'48"

3 [2]

12'00"

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

12'12"

[3]

12'24"

2 1

12'36"

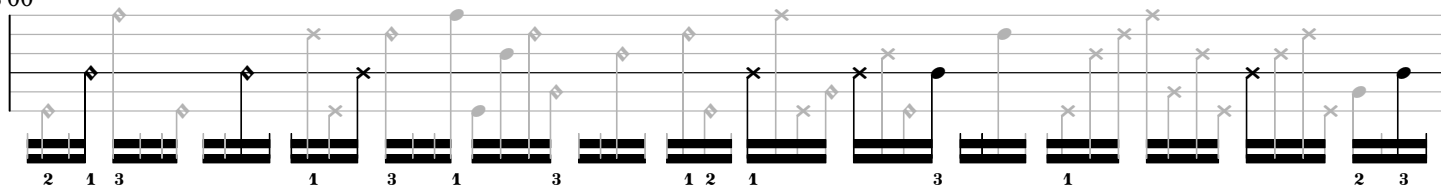
3 1

2 [1] 1

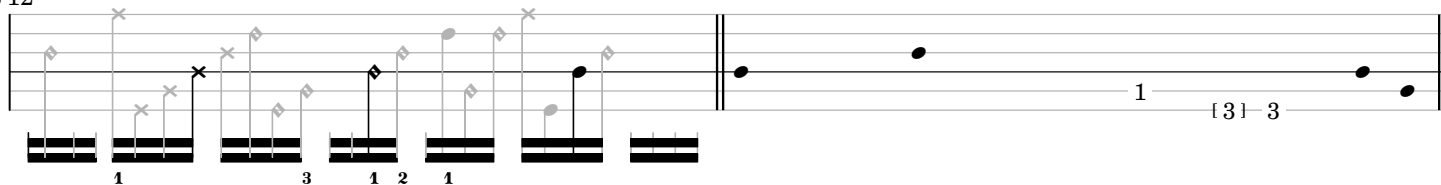
12'48"

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

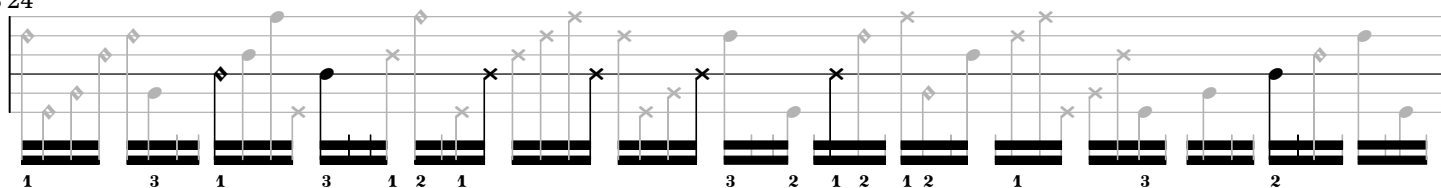
13'00"



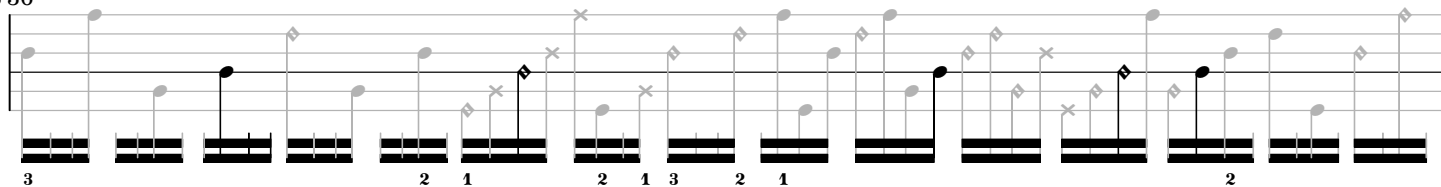
13'12"



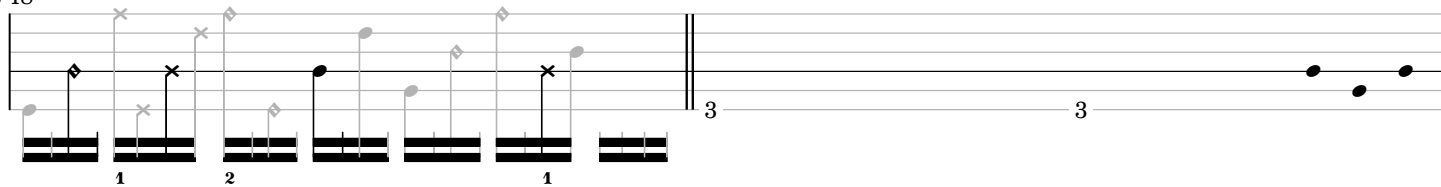
13'24"



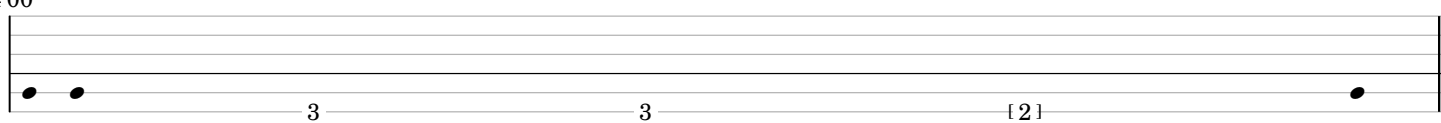
13'36"



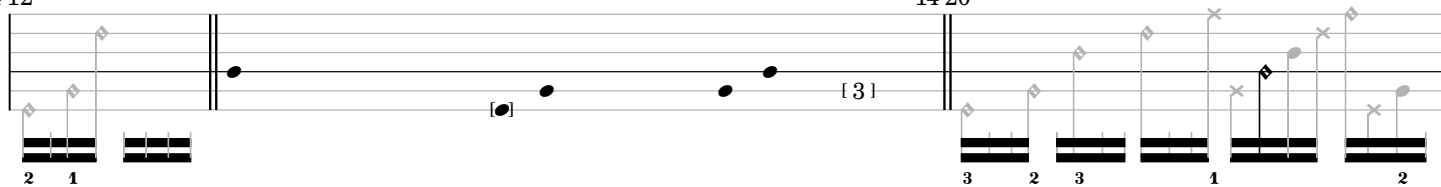
13'48"



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 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]



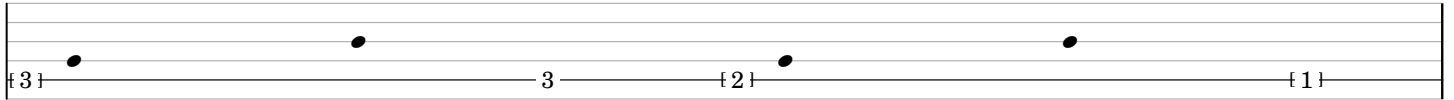
# *ostinato and interrupt*

ensemble part 5

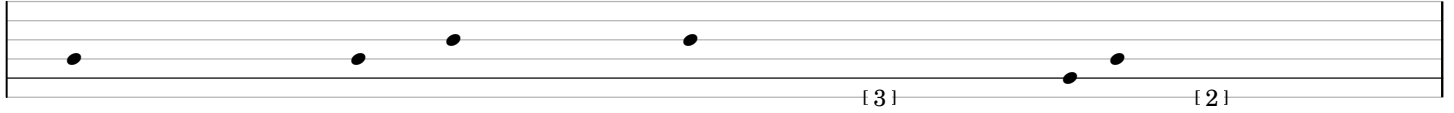
michael winter (mexico city, mx; 2017)

version generated: 2017.08.23

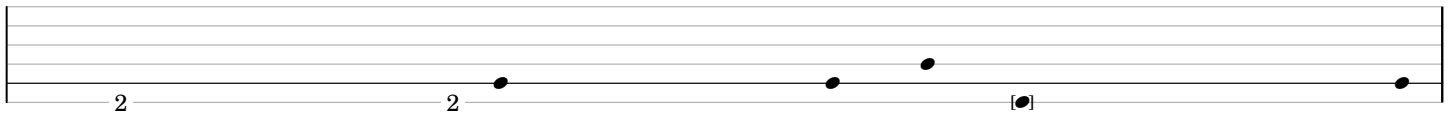
0'12"



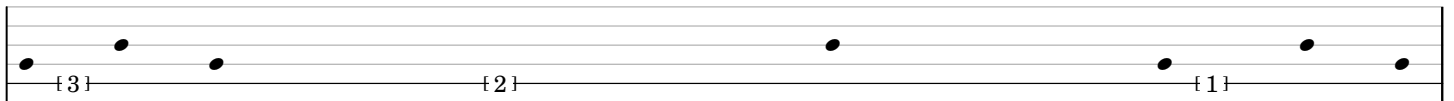
0'24"



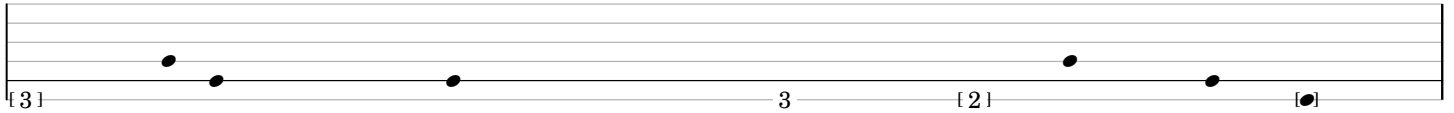
0'36"



0'48"



1'00"

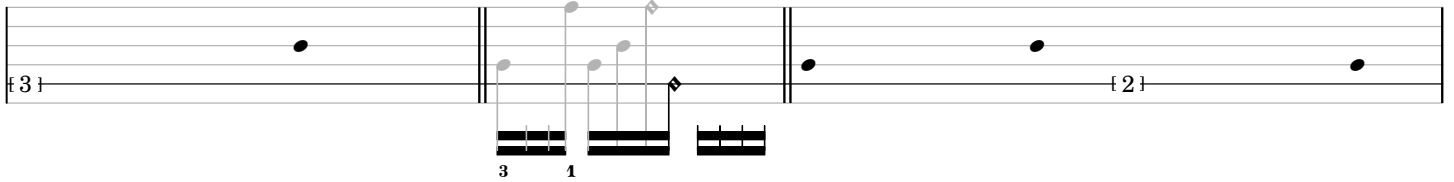


1'12"

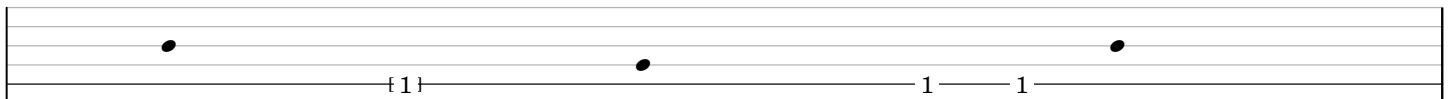


1'24"

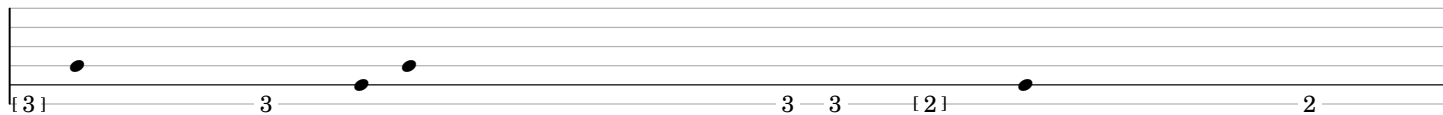
1'28"



1'36"

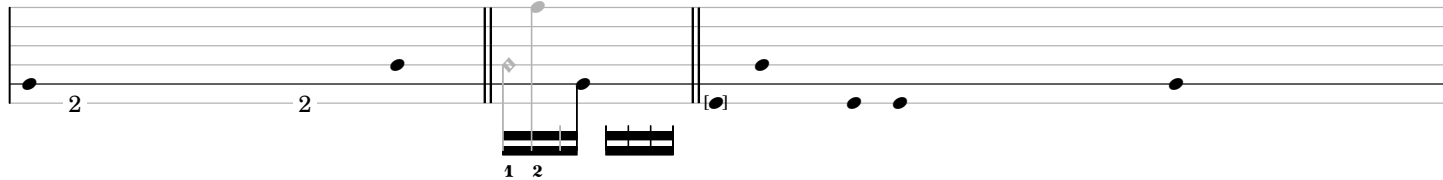


1'48"

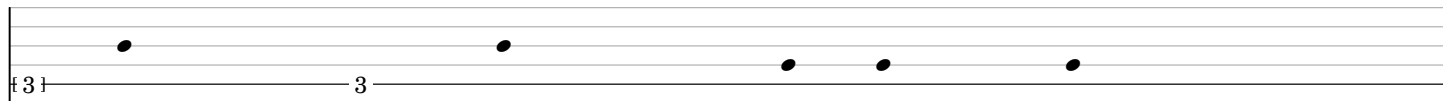


2'00"

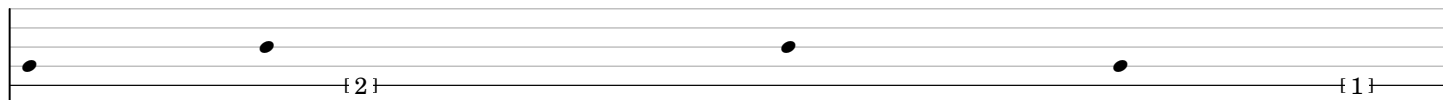
2'04"



2'12"

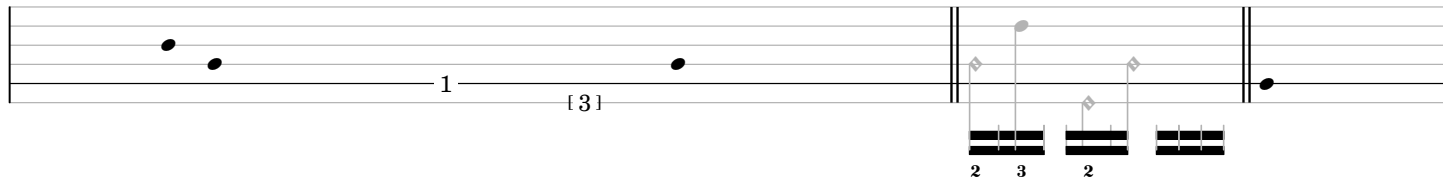


2'24"

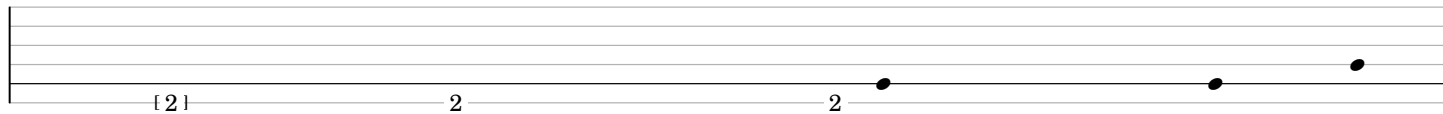


2'36"

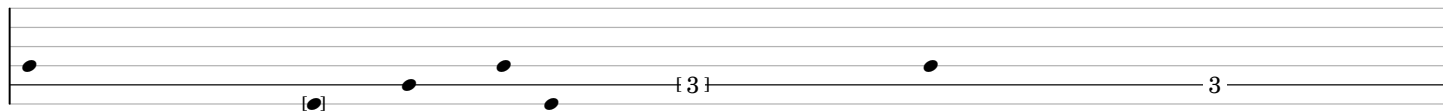
2'44"



2'48"

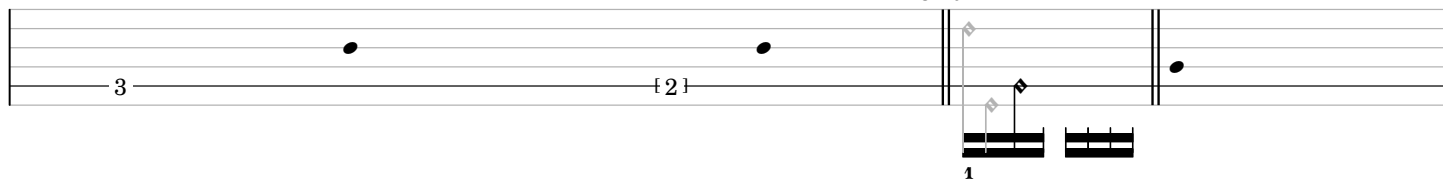


3'00"

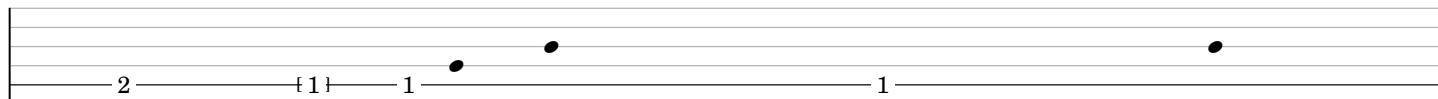


3'12"

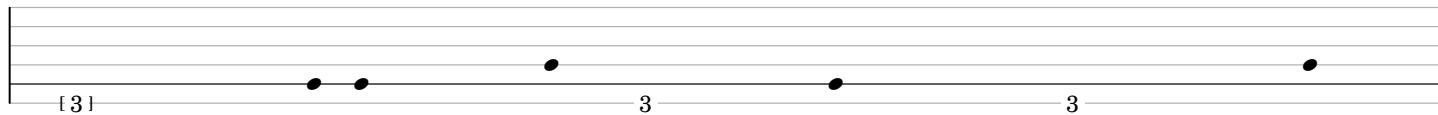
3'20"



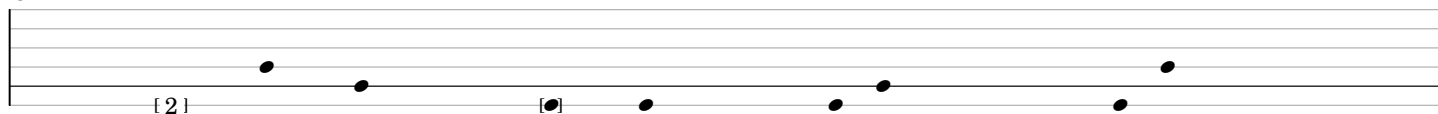
3'24"



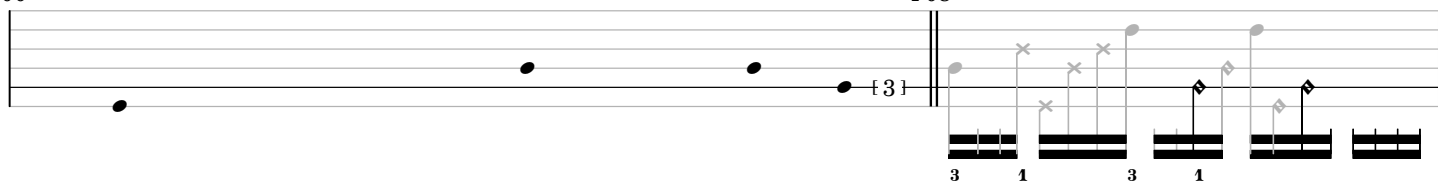
3'36"



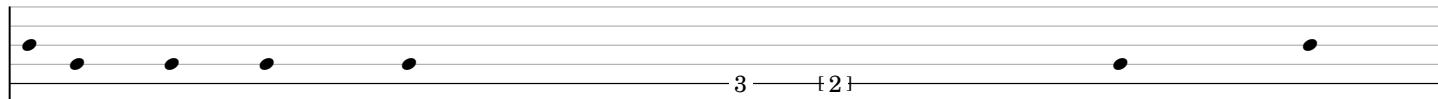
3'48"



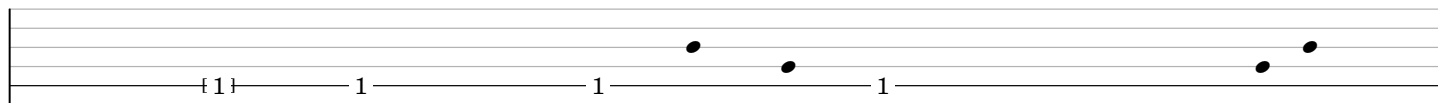
4'00"



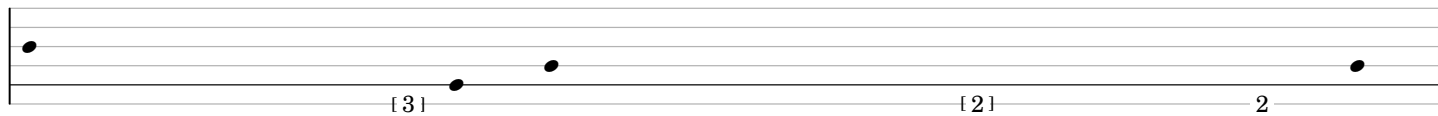
4'12"



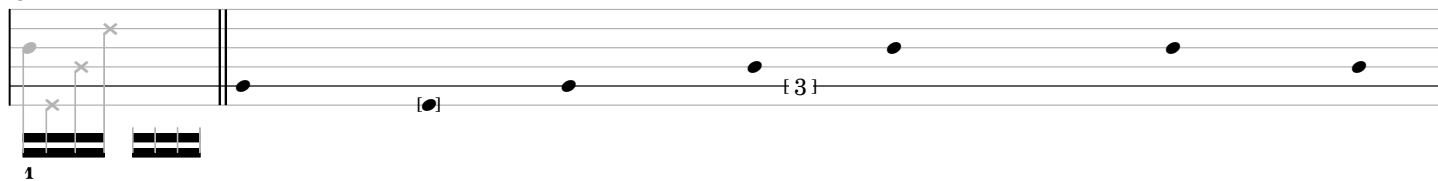
4'24"



4'36"



4'48"

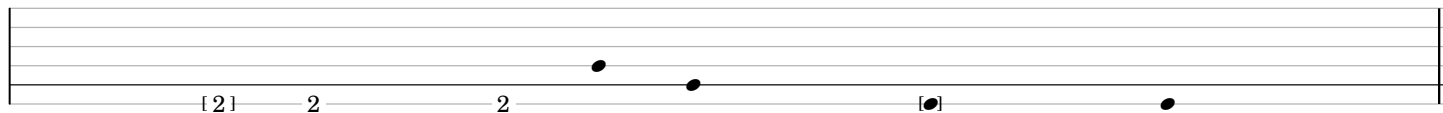


5'48"

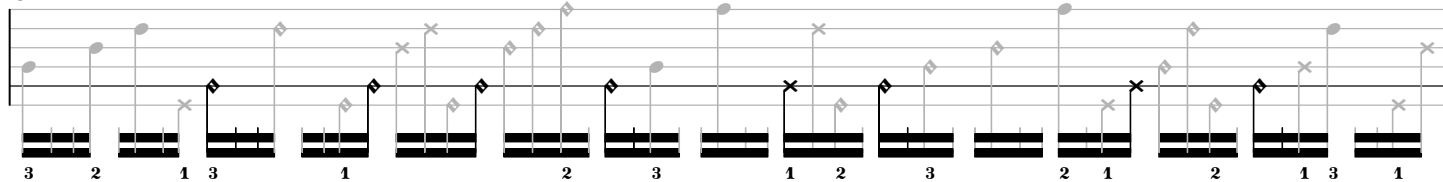
The musical score for '5'48"' consists of a single staff. It begins with a series of sixteenth-note chords, each marked with a '2' below it, indicating a doublet. These are followed by a series of eighth-note chords, each marked with a '1' below it, indicating a single. The score then transitions to a series of whole notes, each marked with a '2' below it, indicating a doublet. The score ends with a double bar line.

The musical score for 'The Great Wall of China' is presented on a grand staff. The piece begins with a 6'12" section, followed by a 6'16" section. The notation includes various musical symbols such as notes, rests, and dynamic markings. The score is divided into measures, with some measures containing multiple notes or rests. The piece concludes with a final measure marked with a double bar line.

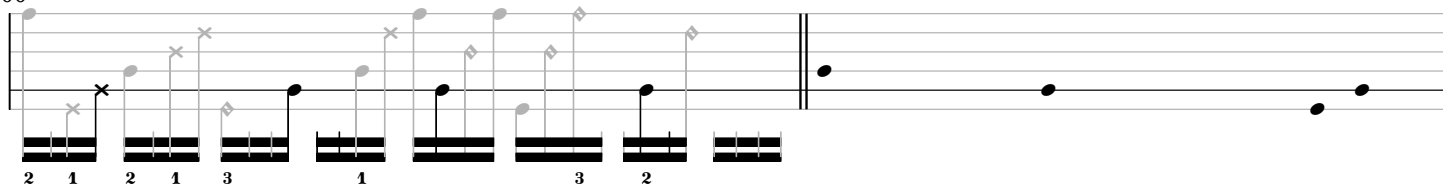
6'36"



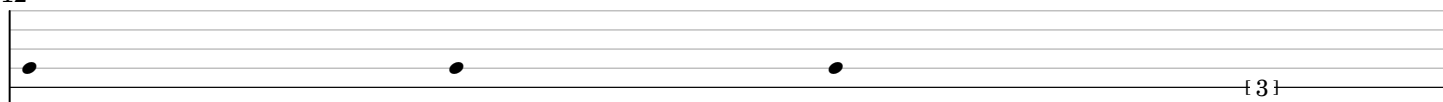
6'48"



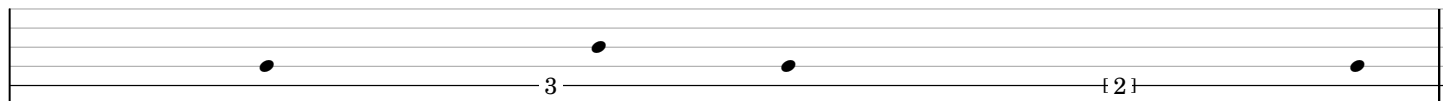
7'00"



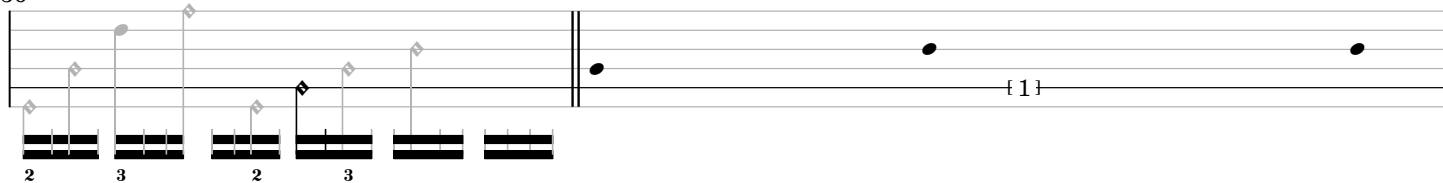
7'12"



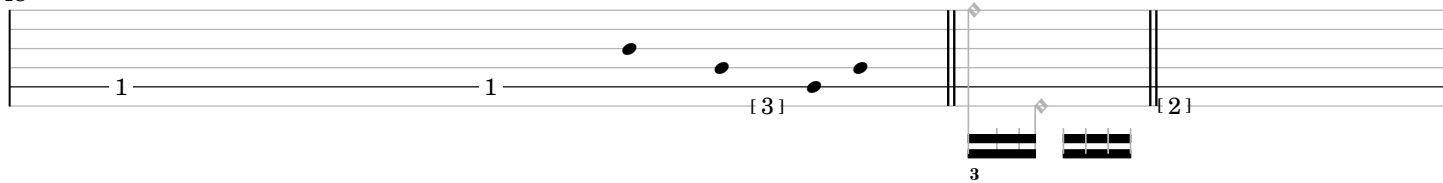
7'24"



7'36"

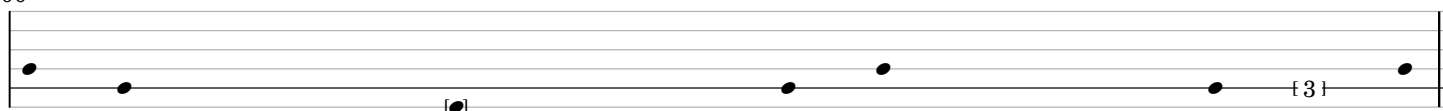


7'48"

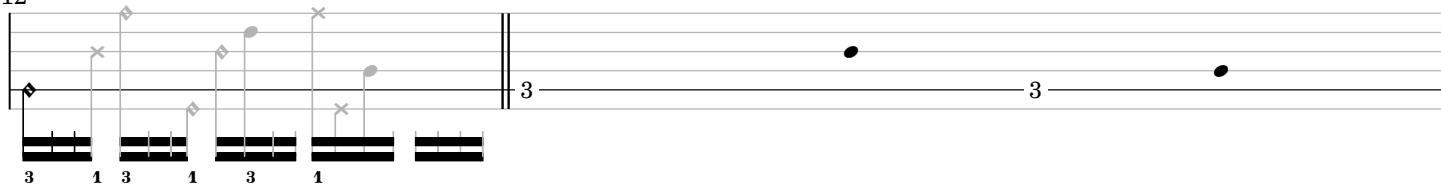


7'56"

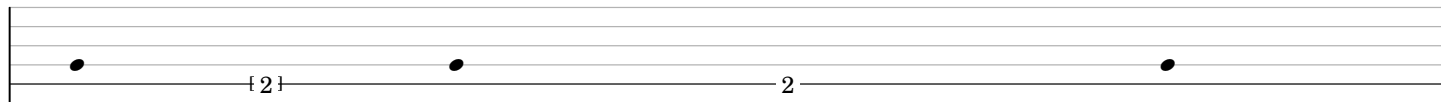
8'00"



8'12"

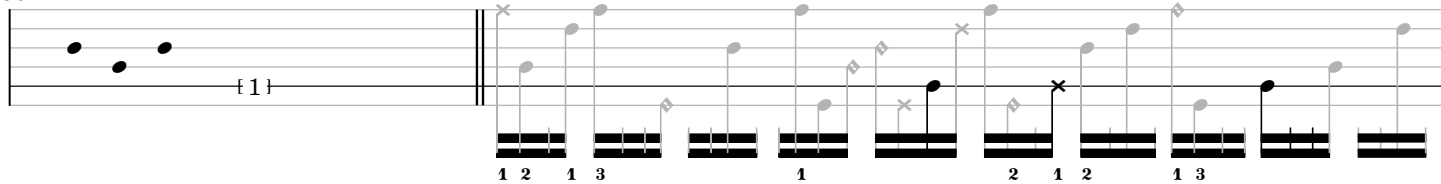


8'24"

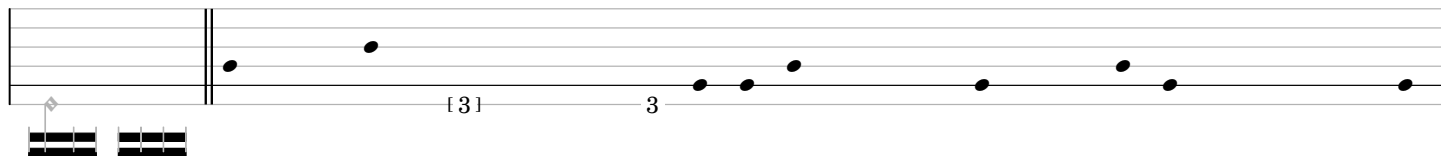


8'36"

8'40"

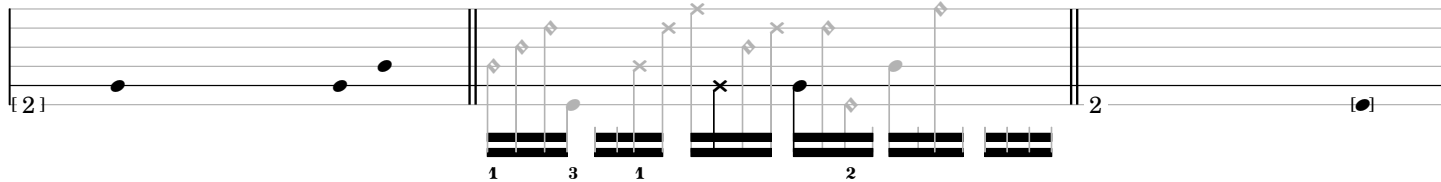


8'48"

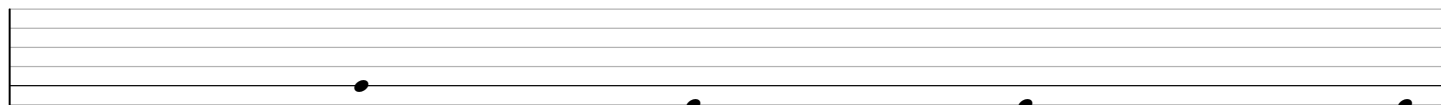


9'00"

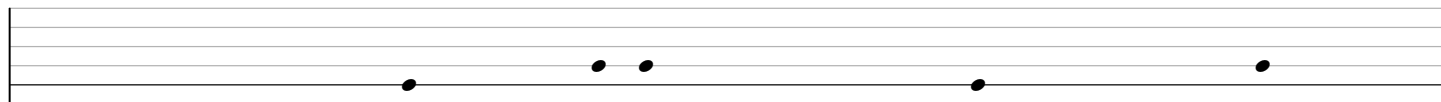
9'04"



9'12"



9'24"

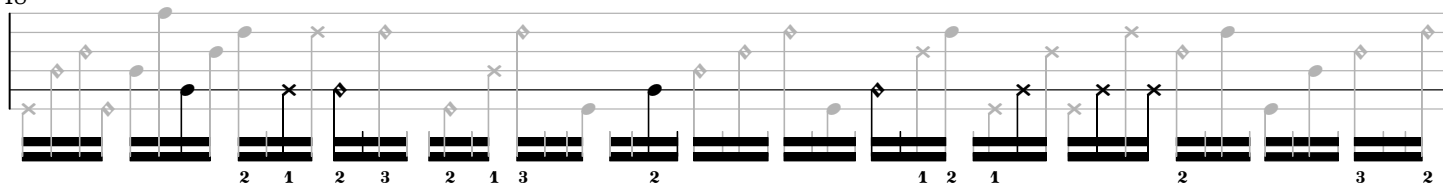


9'36"

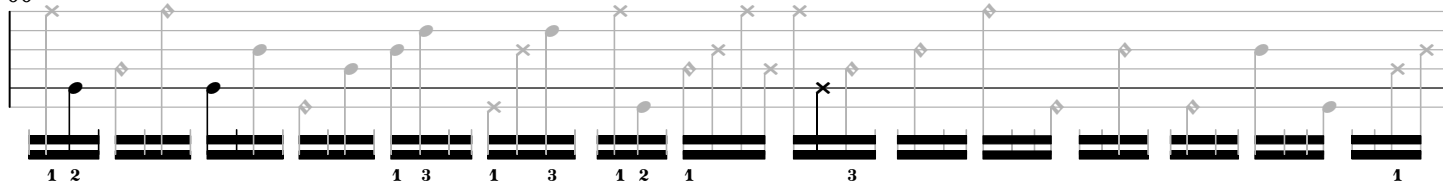
9'44"



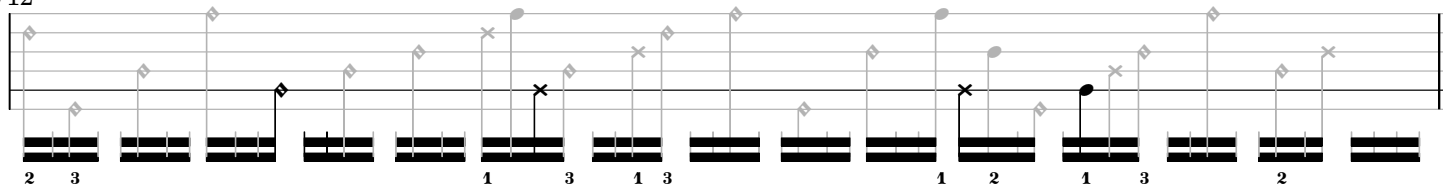
9'48"



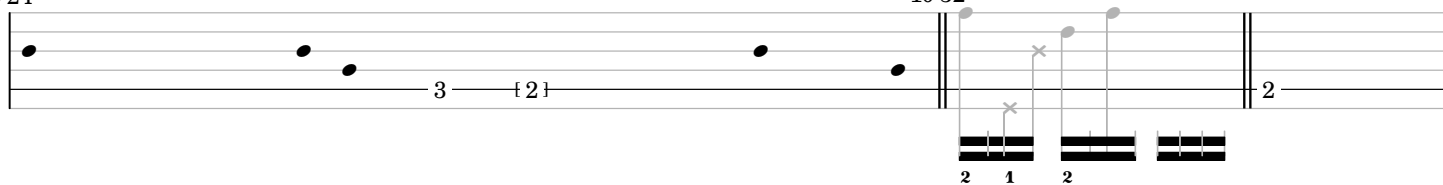
10'00"



10'12"

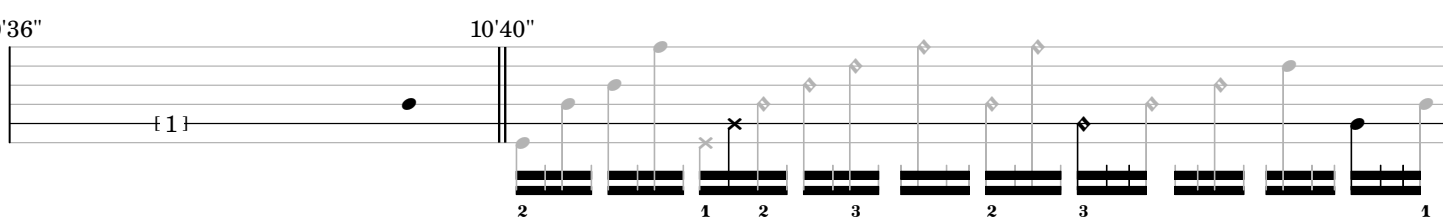


10'24"



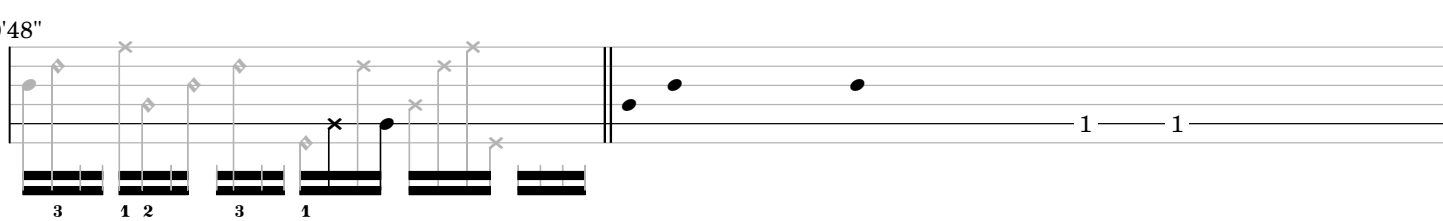
10'32"

10'36"

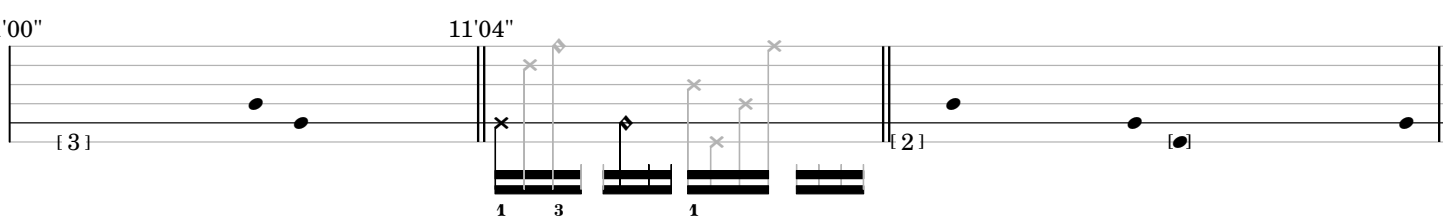


10'40"

10'48"



11'00"



11'04"

11'12"



11'24"

2 1 3 2 3 2 1 3 1 2 1

11'36"

1 2 1

11'44"

1 3 1

11'48"

3 {2}

12'00"

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

12'12"

{3}

12'24"

2 1

12'36"

3 1

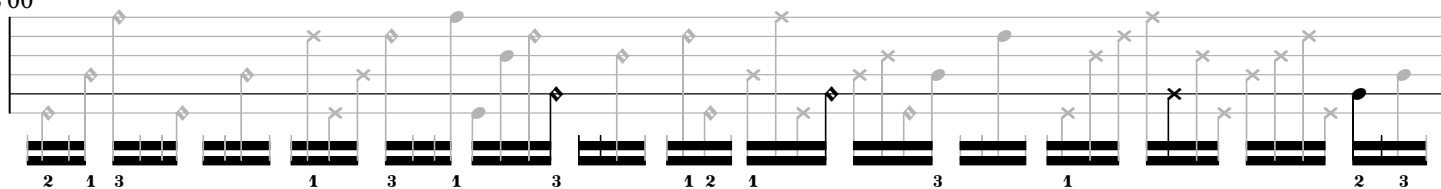
2 {1} 1

12'48"

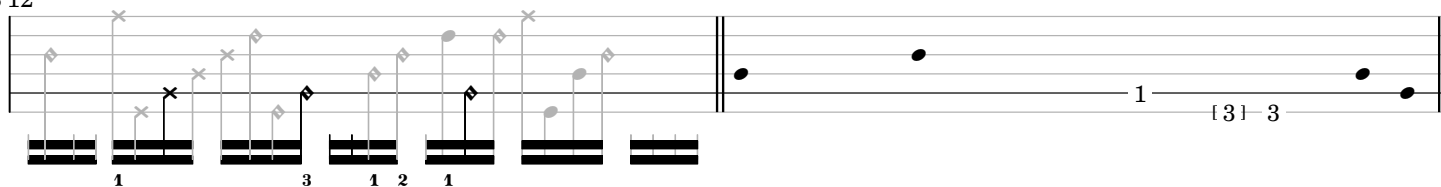
2 3 1 2 1 3 1 3 1 3



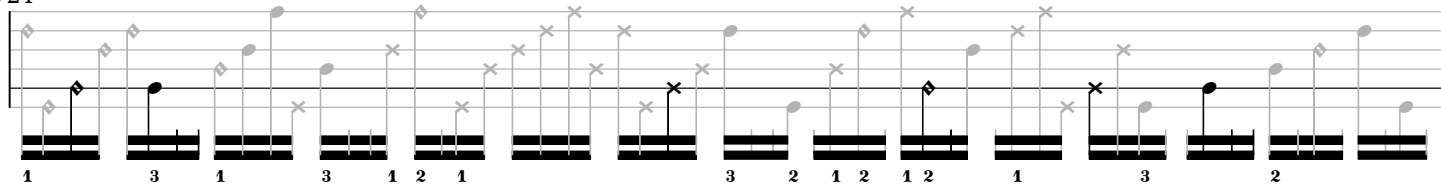
13'00"



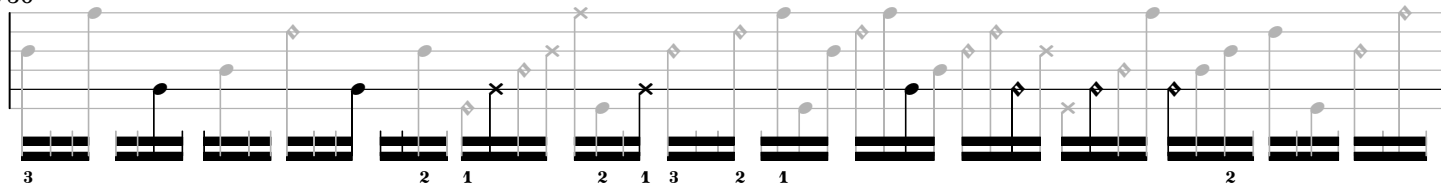
13'12"



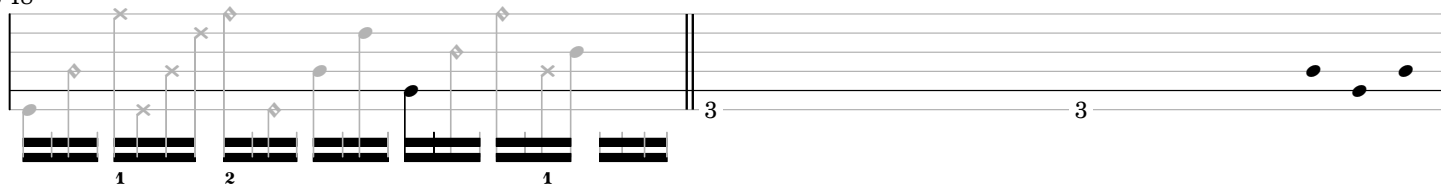
13'24"



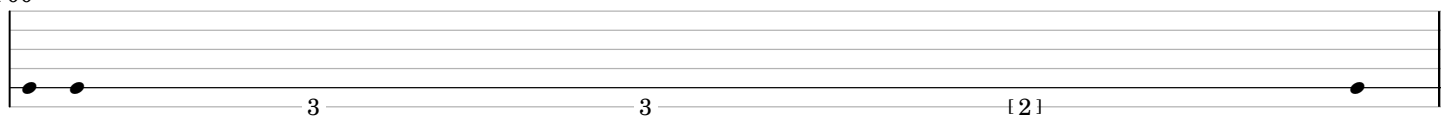
13'36"



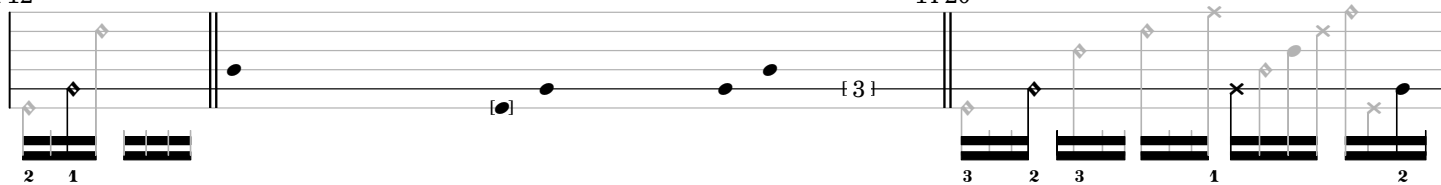
13'48"



14'00"

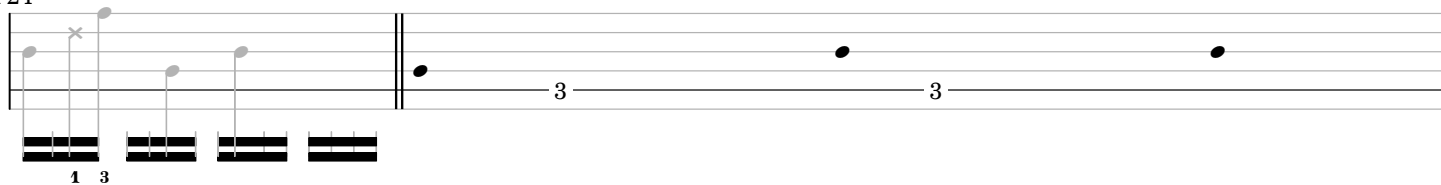


14'12"



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 3 2

15'00"

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

15'36"

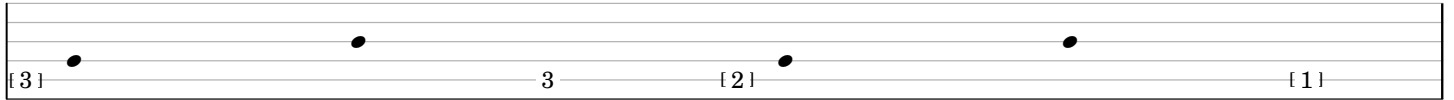
{ 2 } 2 [ 6 ]

# *ostinato and interrupt*

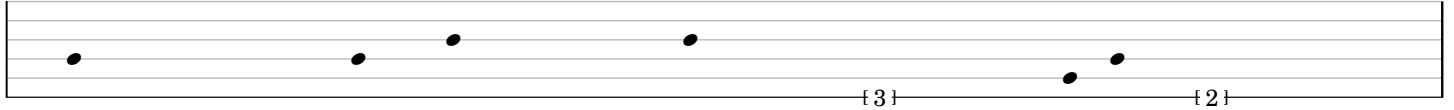
ensemble part 6

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

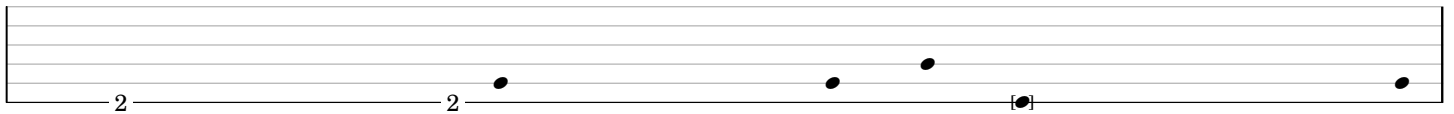
0'12"



0'24"



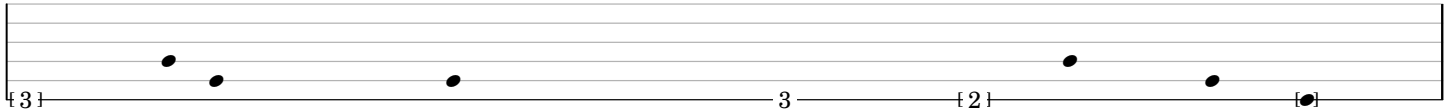
0'36"



0'48"



1'00"

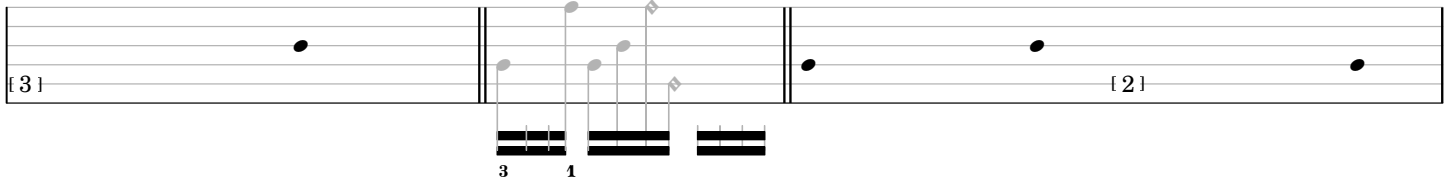


1'12"



1'24"

1'28"



1'36"



1'48"

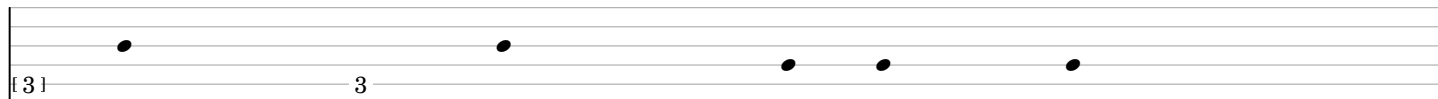


2'00"

2'04"



2'12"

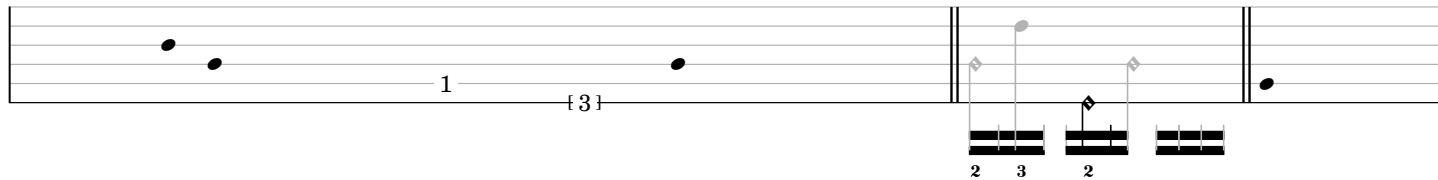


2'24"

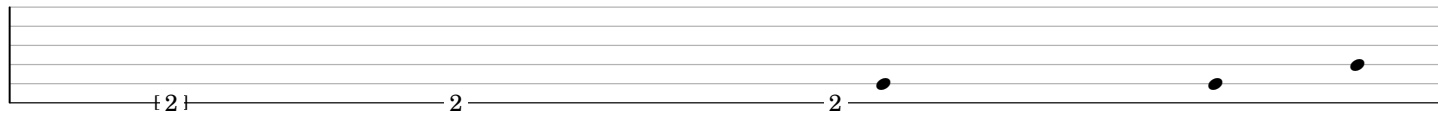


2'36"

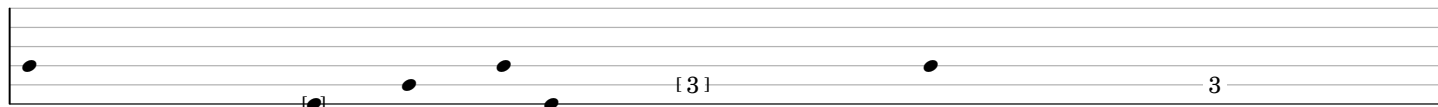
2'44"



2'48"

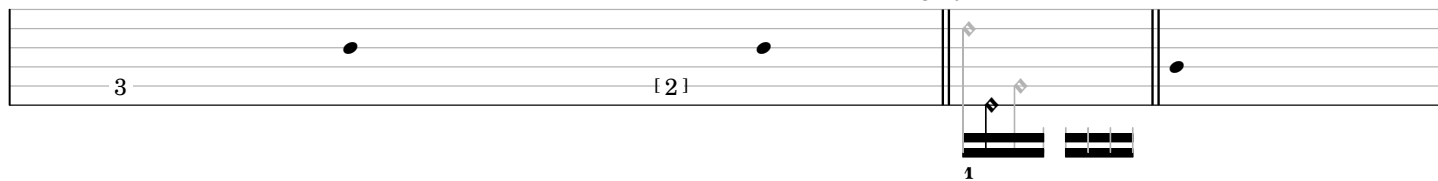


3'00"



3'12"

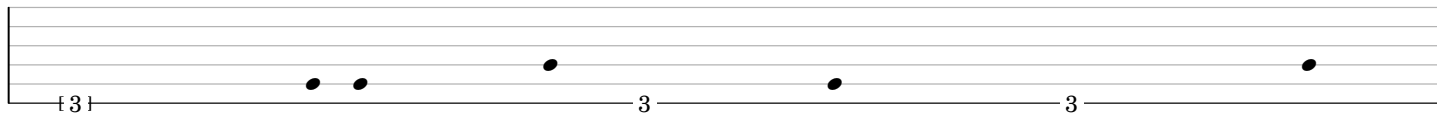
3'20"



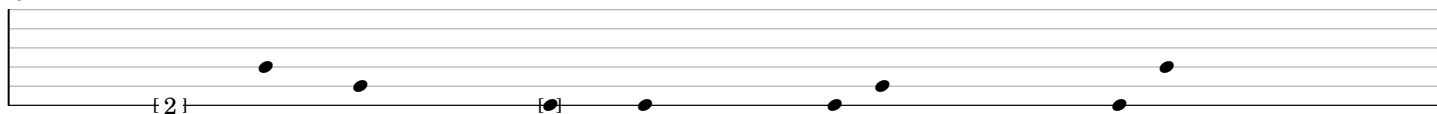
3'24"



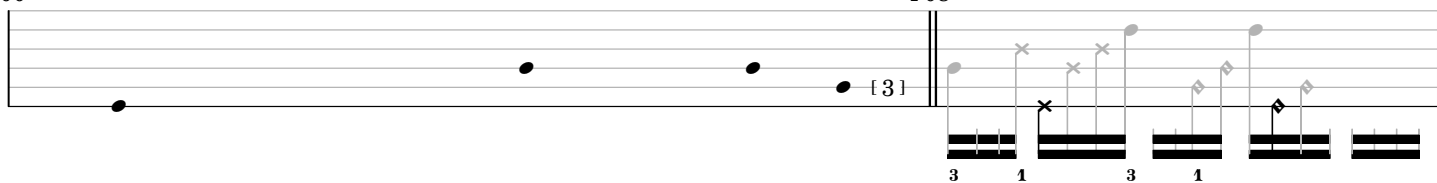
3'36"



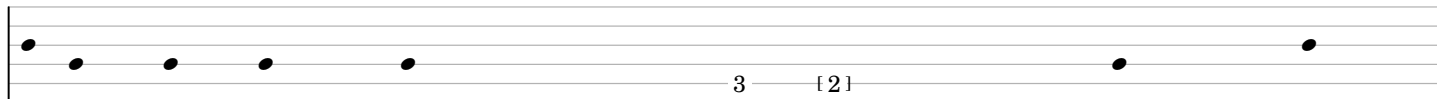
3'48"



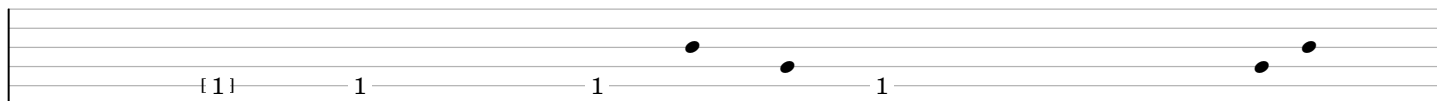
4'00"



4'12"



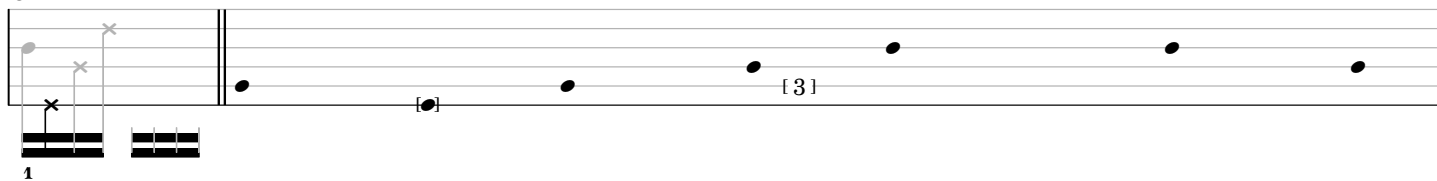
4'24"



4'36"

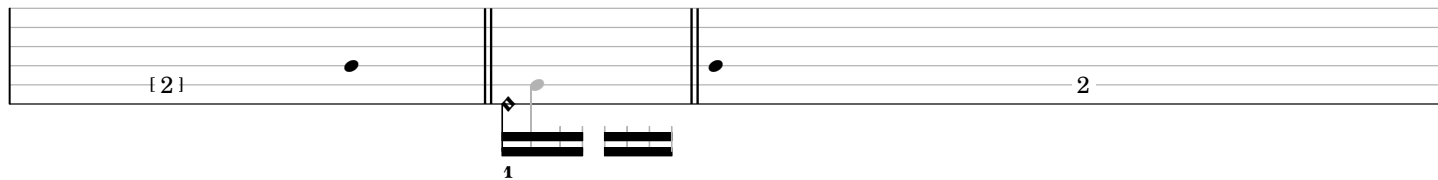


4'48"



5'00"

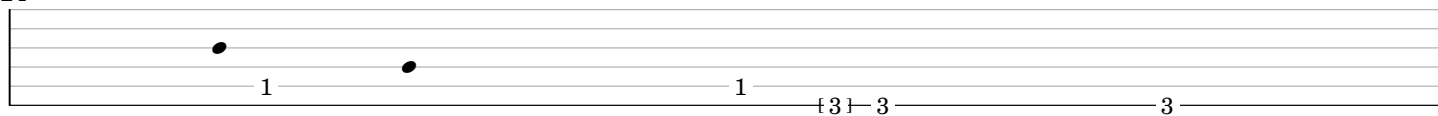
5'04"



5'12"

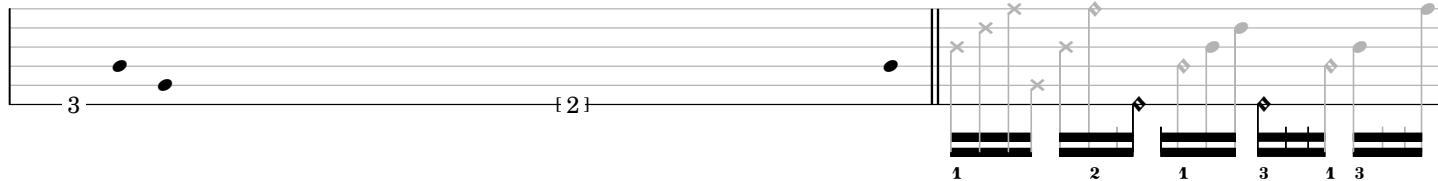


5'24"

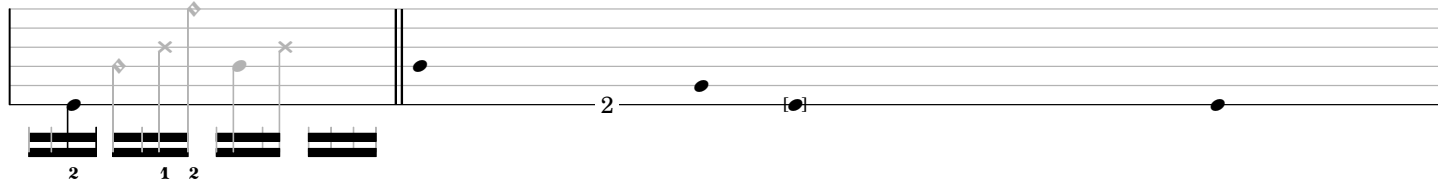


5'36"

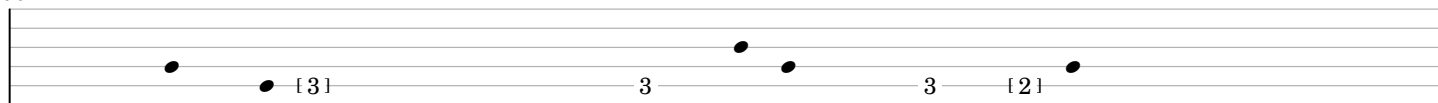
5'44"



5'48"

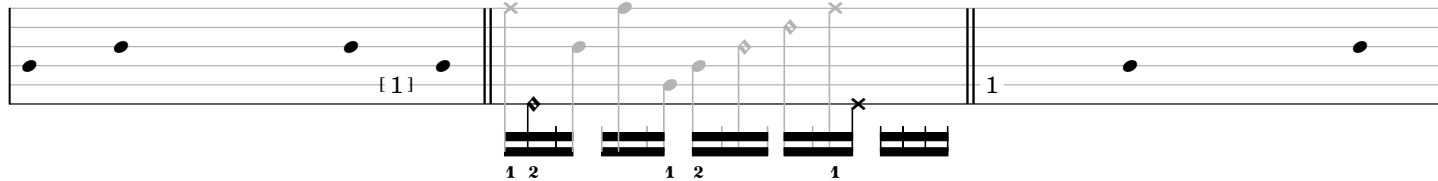


6'00"

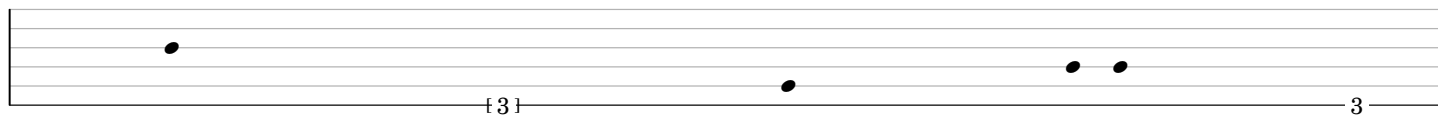


6'12"

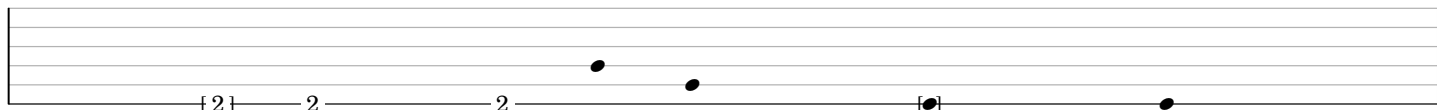
6'16"



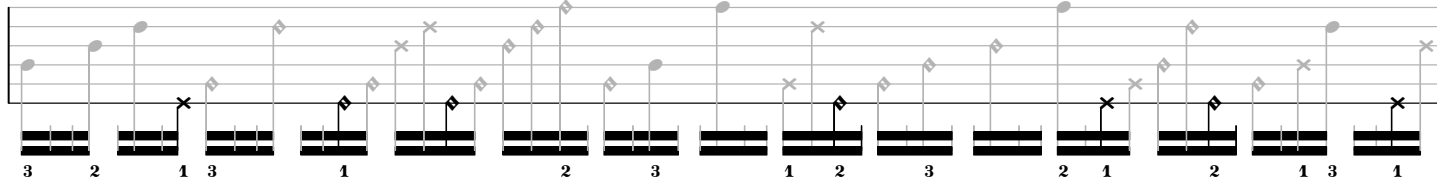
6'24"



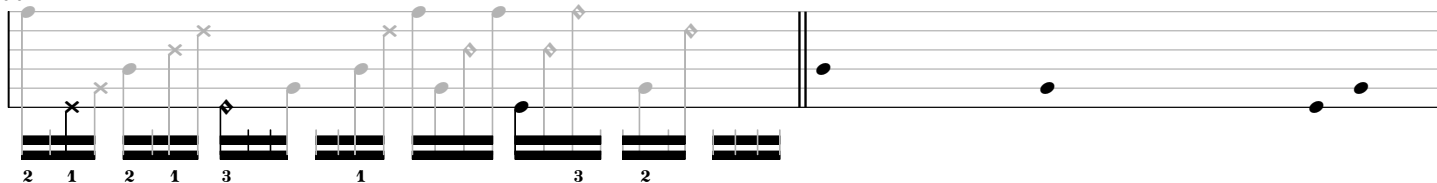
6'36"



6'48"



7'00"



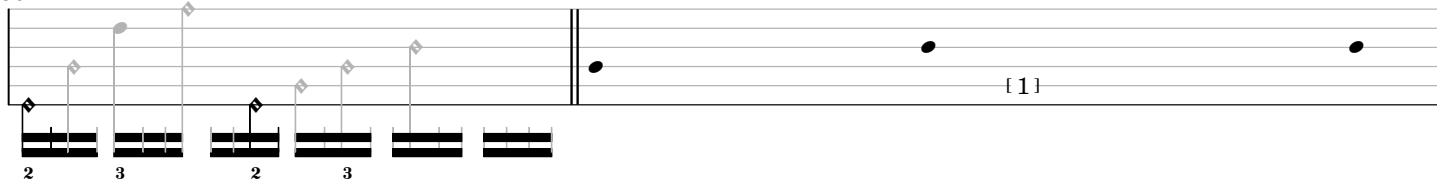
7'12"



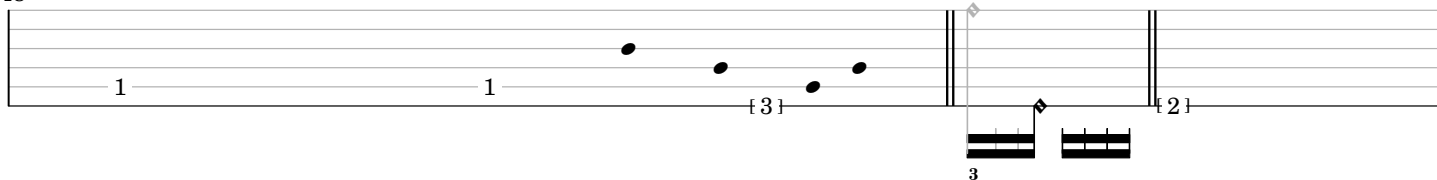
7'24"



7'36"



7'48"



7'56"

8'00"



8'12"

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. A double bar line is present.

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. Below the staff, there are two groups of notes with fingerings 2 and {2}.

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.

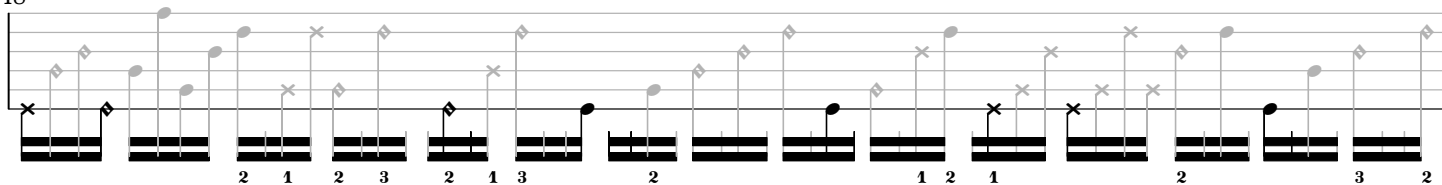
9'36"

9'44"

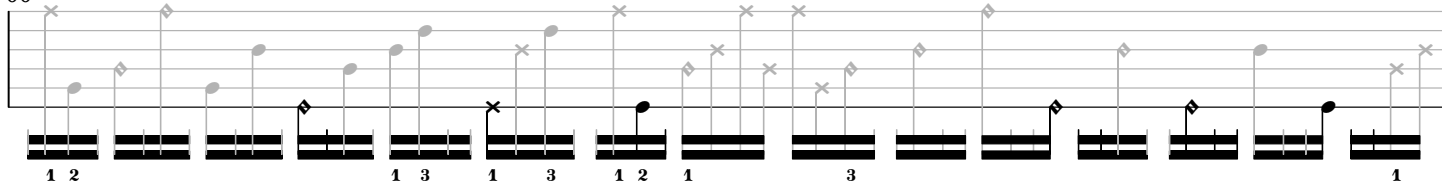
Musical notation for 9'36" and 9'44". The staff shows a sequence of notes with various articulations (diamonds, crosses) and fingerings (3, 1, 2, 1). A double bar line is present. Below the staff, there are two groups of notes with fingerings {3} and 3.



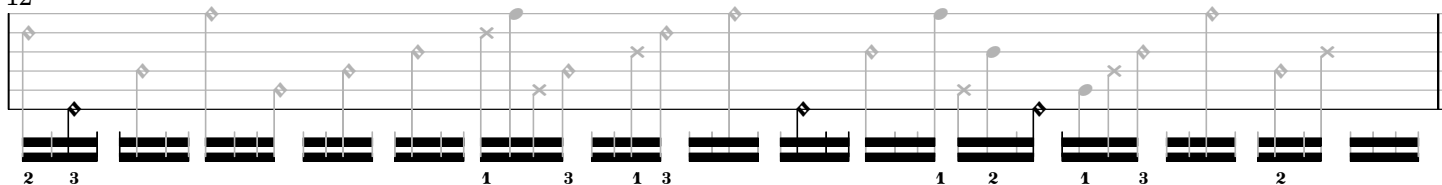
9'48"



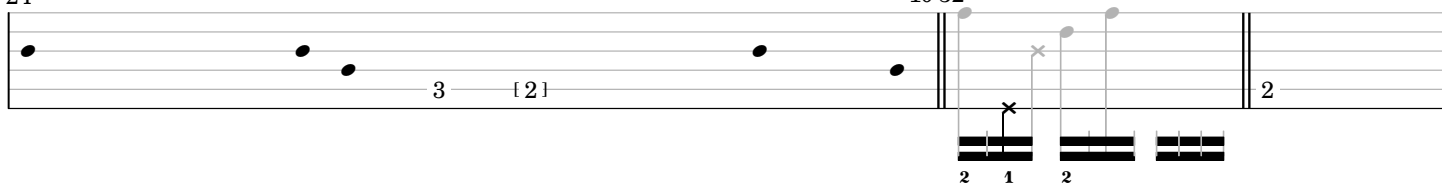
10'00"



10'12"

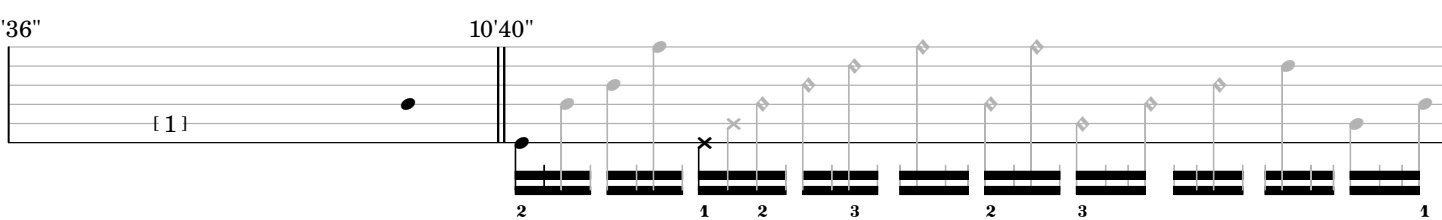


10'24"



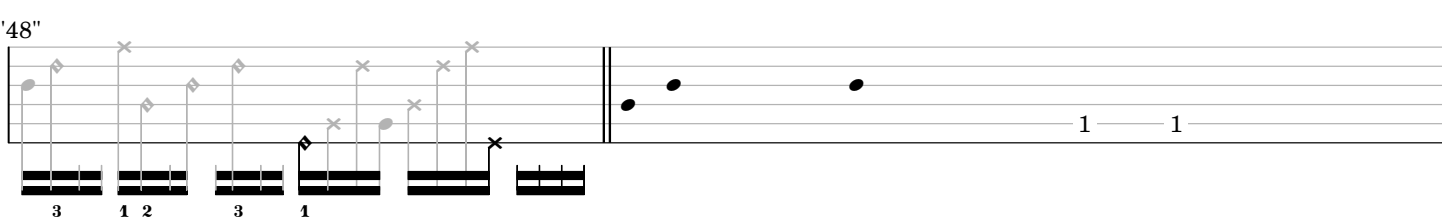
10'32"

10'36"

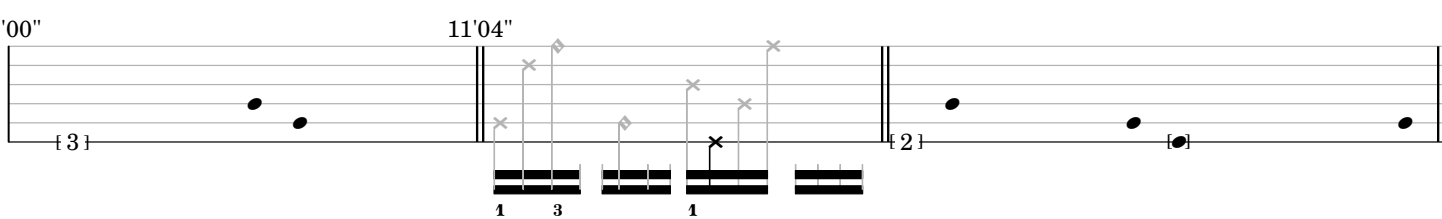


10'40"

10'48"

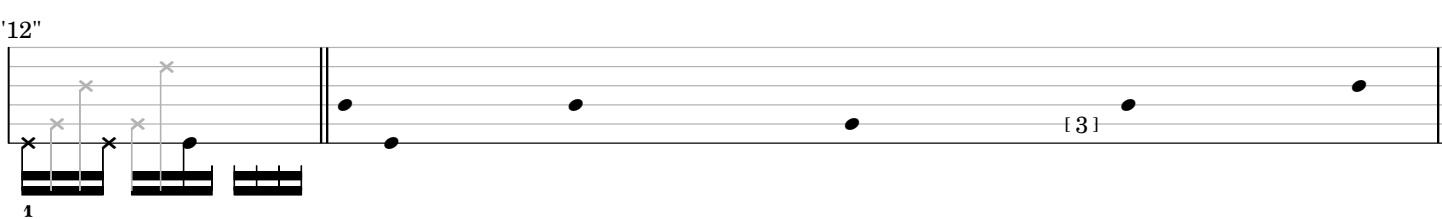


11'00"



11'04"

11'12"



11'24"

The first system of musical notation for 'The Little Boat' is presented on a five-line staff. It begins with a treble clef and a key signature of one flat (B-flat). The melody is written in a simple, stepwise fashion, with notes often beamed in groups of two or three. Fingerings are indicated by numbers 1, 2, and 3 below the notes. There are several accidentals: a B-flat at the beginning, and several B-naturals and A-naturals later in the system. The system concludes with a double bar line, followed by a repeat sign and the number 2, indicating a second ending.

11'36"

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 with notes G4, A4, Bb4, and C5, with various ornaments (x, diamond, cross) and fingerings (1, 2, 1). The lower staff is a bass clef with a key signature of one flat and a 2/4 time signature. It contains a bass line with notes G3, F3, E3, and D3, with various ornaments (x, diamond, cross) and fingerings (1, 2, 1). The system is divided into two measures by a double bar line.

11'44"

11'48"

12'00"

12'12"

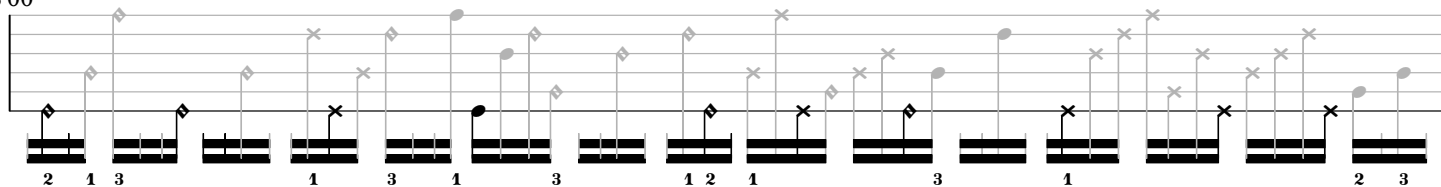
12'24"

The first system of the musical score for 'The Rose Tree' 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 on G4, followed by a quarter note on A4, and then a quarter note on B4. This is followed by a quarter rest, then a quarter note on G4, and a quarter note on A4. The melody continues with a quarter note on B4, a quarter note on A4, and a quarter note on G4. The system ends with a double bar line. Below the staff, there are three groups of notes: the first group has a '2' below it, the second group has a '1' below it, and the third group has no number below it. These groups represent the lyrics 'The', 'Rose', and 'Tree' respectively.

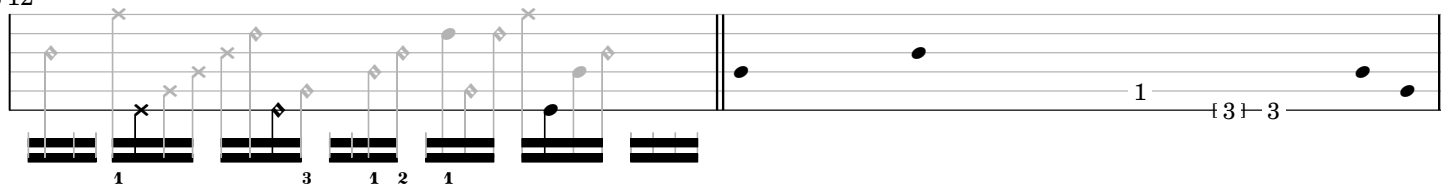
12'36"

12'48"

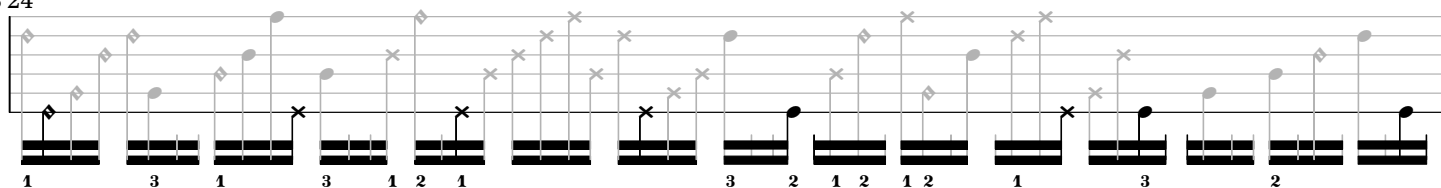
13'00"



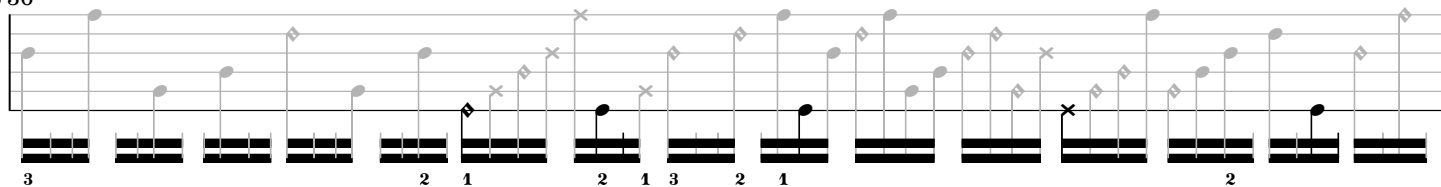
13'12"



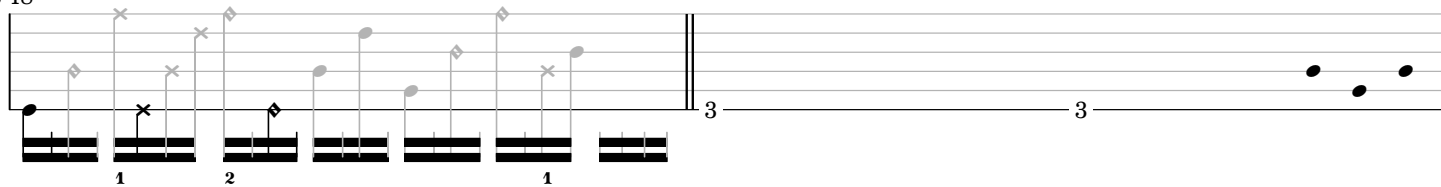
13'24"



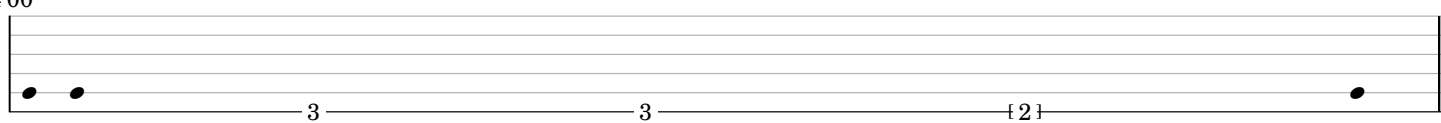
13'36"



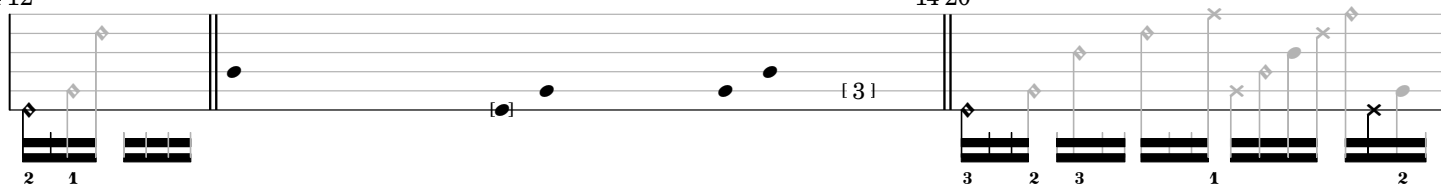
13'48"



14'00"

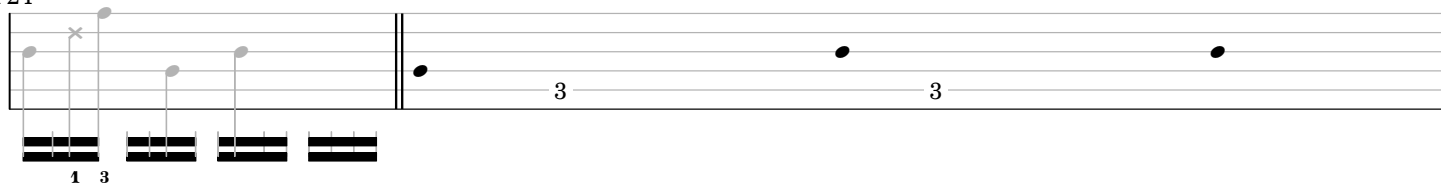


14'12"



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 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}

15'36"

{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    ],

```



```

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
97 // uncomment below to run generator without gui
98 /*
99 "ostinato_and_interrupt_generator_synthdef.scd".loadRelative(true, {
100     ~dir = thisProcess.nowExecutingPath.dirname;
101     ~appStatus = StaticText();
102     ~generateData.value;
103 });
104 */

```

### 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, grey,
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; grey = 70; 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" ++ grey.
84     asString ++ ") " ++ lilyStem ++
85     " \\once \\override NoteHead.color = #(x11-color 'grey" ++ grey.asString ++ ")
86     " ++ lilyNote}));
87
88     for(0, 5, {
89     arg p;
90     parts[p] = parts[p].add(if(string == p || (state == 0),
91     {lilyBar ++ lilyTime ++ lilyStem ++ lilyNote},
92     {lilyBar ++ lilyTime ++ " \\once \\override Stem.color = #(x11-color 'grey" ++ grey.
93     asString ++ ") " ++ lilyStem ++
94     " \\once \\override NoteHead.color = #(x11-color 'grey" ++ grey.
95     asString ++ ") " ++ lilyNote}));
96
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 \\|\\' } >>';
121 outFile.write(inString.replace("<<music>>", outString).replace("piece = " ++ '\\part\\', "piece = " ++ '\\

```

```

111     percussion\");
112     outFile.close;
113
114     for(0, 5, {
115         arg p, staff;
116         outFile = File("~/dir +/ " + "../lilypond/ostinato_and_interrupt.lilypond.ensemble_part-" + (6 - p).
117             asString + ".ly", "w");
118         staff = Array.fill(6, {|i| if(i == (5 - p), {"#f "}, {"(x11-color \'grey" ++ grey.asString ++ " ") "})}).
119             join;
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 )
130
131 // uncomment below generate lilypond files without gui (requires resources to exist)
132 /*
133 ~dir = thisProcess.nowExecutingPath.dirname;
134 ~generateLilypond.value
135 */

```

## 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, 1], masterMuteGroups = #[1, 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)).trunc;
48     SendTrig.kr(Changed.kr(curDur), 0, curDur);
49     imp = Impulse.kr(10);
50     delimp = Delay1.kr(imp);
51     SendReply.kr(imp,
52         '/allMasterLevels',
53         values: [Amplitude.kr(allMaster)]);
54     SendReply.kr(imp,
55         '/groupMasterLevels',
56         values: [
57             Amplitude.kr(guitarOMaster) ++ Amplitude.kr(guitarIMaster) ++
58             Amplitude.kr(percussionMaster) ++ Amplitude.kr(ensembleMaster) ++ Amplitude.kr(accompMaster)];
59     SendReply.kr(imp,
60         '/groupTrackLevels',
61         values: [Amplitude.kr(guitarOTracks) ++ Amplitude.kr(guitarITracks) ++
62             Amplitude.kr(percussionTracks) ++ Amplitude.kr(ensembleTracks) ++ Amplitude.kr(accompTracks)]);
63 }) .add;
64 )

```

```

1  (
2  //~FUNCTION THAT GENERATES THE GUI
3  ^generateGUI = {
4      var win, clockStringFunc, 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];
15     goMute = giMute = pMute = eMute = aMute = [1, 1, 1, 1, 1];
16     goPan = giPan = pPan = ePan = aPan = [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 secInt;
25         var min, sec;
26         sec = secInt;
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
34     ^appStatusFunc = Task({
35         loop {
36             ^appStatus.string = ^appStatusString ++ "*".defer;
37             0.5.wait; {^appStatus.string = ^appStatusString ++ "* *".defer;
38             0.5.wait; {^appStatus.string = ^appStatusString ++ "* * *".defer;
39             0.5.wait; {^appStatus.string = ^appStatusString ++ "* * * *".defer;
40             0.5.wait; {^appStatus.string = ^appStatusString ++ "* * * * *".defer;
41             0.5.wait;
42         }
43     });
44
45     win = Window("ostinato and interrupt", Rect(500, 500, 1100, 500), false).front;
46     masterView = {
47         var view, masterIndicators, master, generator, transport, ranSeed, startPosText, pauseButton, clock;
48
49         OSCFunc({ arg msg, time;
50             {clock.string = clockStringFunc.value(msg[3])}.defer;
51         }, '/tr', s.addr);
52         OSCFunc.new({arg msg; {
53             {i| masterIndicators[i].value = msg[3 + i].ampdb.linlin(-40, 0, 0, 1)} ! 2}.defer;
54         }, '/allMasterLevels', s.addr);
55
56         view = View(win);
57         masterIndicators = [LevelIndicator(), LevelIndicator()];
58         master = HLayout(
59             VLayout(
60                 HLayout(
61                     Slider(view).value.(0.8).action-({|v| ^play.set(\allMasterVol, v.value * 1.25)
62                     },
63                     masterIndicators[0],
64                     masterIndicators[1]),
65                     Button(view).states-([["mute", Color.black], ["mute", Color.black, Color.grey]]).
66                     action-({|v| ^play.set(\allMasterMute, (1 - v.value).abs)}),
67                     StaticText(view).string-("master").align-(\center),
68                     StaticText(view).string-("(all)").align-(\center)),
69                 nil);
70         generator = HLayout(
71             Button(view).states-([["generate"]]).action-({
72                 ^appStatusString = "generating data";
73                 ^appStatusFunc.start;
74                 ^generateData.value(seed: ranSeed.string.asInteger);
75             }),
76             ranSeed = TextField(view).string-("20170121"),
77             Button(view).states-([["reset seed"]]).action-({ ranSeed.string = "20170121"},
78             Button(view).states-([["random seed"]]).action-({ ranSeed.string = 50000000.rand.asString}),
79             [^appStatus = StaticText(view).string-("status: ready"), stretch: 1],
80             nil);
81         transport = HLayout(
82             Button(view).states-([["play", Color.black], ["stop", Color.black, Color.grey]]).action-({| pState |
83                 pauseButton.value = 0;
84                 if(pState.value == 0, {^play.set(\playRate, 0, \startTrig, 0); clock.string =
85                     startPosText.string,
86                     ^play.set(\startPos, startPos, \playRate, 1, \startTrig, 1)})),
87             pauseButton = Button(view).states-([["pause", Color.black], ["pause", Color.black, Color.grey
88             ]]).action-({| pState |
89                 if(pState.value == 1, {^play.set(\playRate, 0)}, {^play.set(\playRate, 1)})),
90             StaticText(view).string-("start time"),

```

```

87         [Slider(view, Rect(0, 0, 30, 20)).action-
88             {|pos|
89                 var min, sec;
90                 startPosText.string = clockStringFunc.value(pos.value * ~totalDur);
91                 startPos = pos.value;
92             }], stretch: 1],
93         startPosText = StaticText(win).string_("00:00").font_(Font("Monaco", 15)),
94         nil);
95     view.layout_(HLayout(master,
96         [VLayout(generator,
97             clock = StaticText(win).string_("00:00").font_(Font("Monaco", 300));,
98             transport
99         ), alignment: \top]))];
100     faderViews = { |group|
101         var view, masterIndicators, trackIndicators, master, tracks;
102         view = View(win);
103         masterIndicators = {LevelIndicator()} ! 10;
104         trackIndicators = {LevelIndicator()} ! 26;
105
106         OSCFunc.new({arg msg; {
107             {|i| masterIndicators[i].value = msg[3 + i].ampdb.linlin(-40, 0, 0, 1)} ! 10}.defer},
108             '/groupMasterLevels', s.addr);
109         OSCFunc.new({arg msg; {
110             {|i| trackIndicators[i].value = msg[3 + i].ampdb.linlin(-40, 0, 0, 1)} ! 26}.defer},
111             '/groupTrackLevels', s.addr);
112
113         master = HLayout(
114             VLayout(
115                 [HLayout(
116                     Slider(view).value_(0.8).action-
117                         {|v| masterVolGroups[group] = v.value * 1.25; ~play.set(
118                             \masterVolGroups, masterVolGroups)},
119                     masterIndicators[group * 2],
120                     masterIndicators[group * 2 + 1]), stretch: 2],
121                     Button(view).states-([["mute", Color.black], ["mute", Color.black, Color.grey]]).
122                         action-
123                             {|v| masterMuteGroups[group] = (1 - v.value).abs; ~play.set(\masterMuteGroups,
124                                 masterMuteGroups)},
125                     StaticText(view).string-("master").align-(\center),
126                     StaticText(view).string-("++groupNames[group]++").align-(\center)
127                 ),
128                 nil);
129         tracks = { |part|
130             HLayout(
131                 VLayout(
132                     HLayout(
133                         Slider(view).value_(0.8).action-
134                             {|v| volGroups[group][part] = v.value * 1.25; ~play.set(
135                                 groupAbbr[group] ++ "Vol", volGroups[group])},
136                             trackIndicators[group * 6 + part - if(group > 0, {2}, {0})],
137                             Button(view).states-([["mute", Color.black], ["mute", Color.black, Color.grey
138                                 ]]).action-
139                                 {|v| muteGroups[group][part] = (1 - v.value).abs; ~play.set(groupAbbr[
140                                     group] ++ "Mute", muteGroups[group])},
141                             StaticText(view).string-("pan").align-(\center),
142                             Knob(view).value_(0.5).action-
143                                 {|v| panGroups[group][part] = v.value * 2 - 1; ~play.set(groupAbbr[
144                                     group] ++ "Pan", panGroups[group])},
145                             StaticText(view).string-
146                                 if(group == 4, {accompNames[part]}, {(6 - part).asString}).align-
147                                     (\center)
148                         ),
149                         nil)
150                     } ! if((group == 0) || (group == 4), {4}, {6});
151                 view.layout_(HLayout(master, nil, *tracks)) ! 5;
152             tabButtonReset = {masterButton.value = 1;
153                 guitarOButton.value = 1; guitarIButton.value = 1; percButton.value = 1; ensembleButton.value = 1;
154                 accompButton.value = 1;};
155             win.layout = VLayout(
156                 HLayout(
157                     masterButton = Button().states-([["master controls", Color.white, Color.grey], ["master
158                         controls", Color.black]]).action-
159                         {tabButtonReset.value; masterButton.value = 0; tabs.index = 0 }.value(0),
160                     guitarOButton = Button().states-([["guitar (ostinato)", Color.white, Color.grey], ["guitar (
161                         ostinato)", Color.black]]).action-
162                         {tabButtonReset.value; guitarOButton.value = 0; tabs.index = 1 }.value(1),
163                     guitarIButton = Button().states-([["guitar (interrupt)", Color.white, Color.grey], ["guitar (
164                         interrupt)", Color.black]]).action-
165                         {tabButtonReset.value; guitarIButton.value = 0; tabs.index = 2 }.value(1),
166                     percButton = Button().states-([["percussion", Color.white, Color.grey], ["percussion", Color.
167                         black]]).action-
168                         {tabButtonReset.value; percButton.value = 0; tabs.index = 3 }.value(1),
169                     ensembleButton = Button().states-([["interrupt highlights", Color.white, Color.grey], ["
170                         interrupt highlights", Color.black]]).action-
171                         {tabButtonReset.value; ensembleButton.value = 0; tabs.index = 4 }.value(1),
172                     accompButton = Button().states-([["fields / beats / flicker", Color.white, Color.grey],
173                         ["fields / beats / flicker", Color.black]]).action-
174                         {tabButtonReset.value; accompButton.value = 0; tabs.index = 5 }.value(1)),
175                     tabs = StackLayout(masterView.value, faderViews[0], faderViews[1], faderViews[2], faderViews[3],
176                         faderViews[4]);
177                 };
178             )
179         )

```

```

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                                line-stencil)
136                            (* position staff-space 0.5) Y)))
137                    (and (pair? colors)
138                        (set! colors (cdr colors)))))
139                  line-positions)
140      (let* ((line-count (ly:grob-property grob 'line-count 5))
141             (height (* (1- line-count) (/ staff-space 2))))
142        (do ((i 0 (1+ i)))
143            ((= i line-count))
144            (let ((color (if (and (pair? colors)
145                                 (> (length colors) i))
146                        (list-ref colors i)
147                        #f)))
148              (set! total-lines (ly:stencil-add
149                                total-lines
150                                (ly:stencil-translate-axis
151                                 (if (color? color)
152                                     (ly:stencil-in-color line-stencil
153                                                                (first color)
154                                                                (second color)
155                                                                (third color))
156                                     line-stencil)
157                                (- height (* i staff-space) Y))))))
158      total-lines))
159
160  #(define-public (bracket-stencils grob)
161    (let ((lp (grob-interpret-markup grob (markup #:fontsize 3.5 #:translate (cons -0.3 -0.5) "[")))
162          (rp (grob-interpret-markup grob (markup #:fontsize 3.5 #:translate (cons -0.3 -0.5) "]"))))
163      (list lp rp)))
164
165  bracketify = #(define-music-function (parser loc arg) (ly:music?)
166    (.i "Tag @var{arg} to be parenthesized.")
167    #{
168      \once \override ParenthesesItem.stencils = #bracket-stencils
169      \parenthesize $arg
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  }

```