

# User Guide



U.S. Pat. 10,446,126

© XJ Music Inc

[xjmusic.com](http://xjmusic.com)

Table of Contents

<b>Welcome</b>	<b>4</b>
<b>What is XJ music?</b>	<b>5</b>
<b>How It Works</b>	<b>6</b>
<b>Installing the XJ music workstation</b>	<b>7</b>
Windows Installation	7
Mac Installation	7
Linux Installation	8
<b>Getting Started in the Workstation</b>	<b>8</b>
Sign in	9
Libraries	10
Programs	11
Types of Programs	12
Configuration of Programs	13
Instruments	14
Types of Instruments	15
Modes of Instruments	15
Instrument Type-Modes	16
Hook-type Instruments (Loop-mode only)	16
Event-mode Instruments	16
Chord-mode Instruments	16
Performance-mode Instrument	17
Configuration of Instruments	18
Instrument Audio Quantization	18
Templates	19
Binding Template Content	20
Configuration of Templates	21
Preview Templates	1
Fabrication & Playback	1
Ship Keys	1
Fabrication of Segments	1
Dubbing of Output Audio	1
<b>Program Editor</b>	<b>1</b>
Editing a Macro Program	1
Editing a Main Program	1
Editing a Beat Program	1
Editing a Detail Program	1
Instrument Editor	1
<b>Making XJ music</b>	<b>1</b>
Memes Influence Choice	1

Pick a Marble from the Bag	1
Types of Memes	1
Meme Taxonomy	1
<b>Instrument Behavior</b>	<b>1</b>
Events	1
Drum Beat Events	1
Drum Detail Events	1
Percussion Loops	1
Loop Priority	1
Background Ambience	1
Arcs	1
<b>Appendix A. Chord Interpretation</b>	<b>1</b>
<b>Appendix B. Entities &amp; their Relationships</b>	<b>1</b>
Account Model	1
Program Model	1
Instrument Model	1
Segment Model	1
<b>Appendix C. Technical Architecture</b>	<b>1</b>
Backend Services	1
Telemetry	1
<b>Appendix D. Software Operation</b>	<b>1</b>
<b>Glossary</b>	<b>1</b>
<b>References</b>	<b>1</b>
<b>Authors</b>	<b>1</b>

# Welcome

We've had this vision of a completely new approach to music recording and listening, where artists collaborate on pieces stored in a central database, assisted by an algorithmic medium.

That's XJ music: a recording medium with infinite possibilities.

Thousands of hours of work yield infinite hours of listening. Artists collaborate continuously on the fly. The music is never "done" because the music is a breathing, always-changing, never-repeating live stream.

We believe music is fluid and dynamic between spaces and across time.

Songs have been transmitted verbally from person to person for many thousands of years.

The foundation of a song is something more profound than its recording. The combination of melody, harmony, and rhythm transcends language and culture. And we may render each song in infinite ways.

The performer brings sheet music to life like a storyteller making an ancient tale relevant again and again for the contemporary audience.

XJ is an evolution of recorded music in space and time for listeners and artists.

Welcome to XJ music!

**Charney Kaye**

Founder & CEO

# What is XJ music?

XJ is founded on the core principle that ***music is human***. Yet, we seek to innovate what's possible.

We're not playing records back here. You're listening to the future. XJ is a patented system, a new algorithmic medium. We are building a new school of cutting-edge musicians putting this system to work.

Each XJ music channel is a living, breathing organism, constantly changing and never repeating. We work tirelessly on the XJ music platform to cultivate a compelling world for artists in our new collaborative medium. In [the XJ music Podcast](#), the XJ team shares highlights from our journey.

We see an increasing wave of companies promising generative AI-driven music, especially those trying to package and sell their generative music as productivity and wellness. But that's just a line to impress venture capitalists. These companies need to put in the proper effort to understand music.

What we've done here is create an instrument that allows human beings to make generative music and realize some new possibilities based on their existing ideas. There's a barrier to entry to making music with XJ. This technology doesn't make the average person into a musician, but it does allow the experienced musician to express a musical recording across time and space in previously inconceivable ways.

XJ music is ultra-specialized technology. This not something the average person will operate. Artists working in ambient music may be interested in using XJ music technology to produce records.

We're anticipating the most exciting use cases to be in the area of streaming background music, video gaming background music, and environmental music and in spaces.

# How It Works

XJ utilizes a versatile template that accepts various inputs, such as content, programs, and instruments from many libraries. The core editing of this content takes place in a virtual workspace known as the 'Lab.'

XJ's 'play' functionality is more complex than it seems, as it is powered by a feature termed the 'Nexus.' The Nexus meticulously reads the content and constructs an ongoing sequence of segments. Each segment, unique in its timeline placement, consists of all the content employed to forge it.

Creating a segment requires several elements. Initially, you will need 'Macro Programs,' which are comparable to the sequence of songs a DJ would pick to play one after another. Following this, you will need 'Main Programs' - these are analogous to the individual pieces in a DJ's set, complete with chord progressions and specific voicings for each instrument in the song.

As the XJ moves through the creation process, it seeks out 'Drumbeat Programs.' These programs house a collection of drum events that construct the underlying rhythm of the piece. XJ further scours for 'Detail Programs' that execute the voicings derived from the Main Programs, such as a bassline or stab pattern.

To fully realize the sonic details in the programs, XJ will search for the appropriate instruments, each packed with audio content. For instance, a bass detail pattern from the Detail Program merges with bass voicings from the Main Program. This combination then aligns with a polyphonic bass-type instrument selected to curate and choose the final bass audio for the segment.

Some types of instrument audio are utilized formulaically within XJ, such as transition audio, percussion audio, and ambient audio. This creates a cohesive, dynamic, and robust sonic experience. Hence, XJ combines all these elements to generate a beautifully organized and harmonious audio segment, providing a user-friendly, intricate, and comprehensive approach to music creation.

# Installing the XJ music workstation

Download the installation files from this [Google Shared Drive](#)

## Windows Installation

- Download the XJ music workstation Windows installer ZIP file to your chosen directory.
- Extract the contents of the ZIP file with your extraction tool of choice.
- Double-click the Application file in the folder to begin installation.
- Windows Defender may stop the installation process- select More Info and then Run Anyway.
- Follow the steps in the Setup Wizard to choose where you want the workstation to be installed.

## Mac Installation

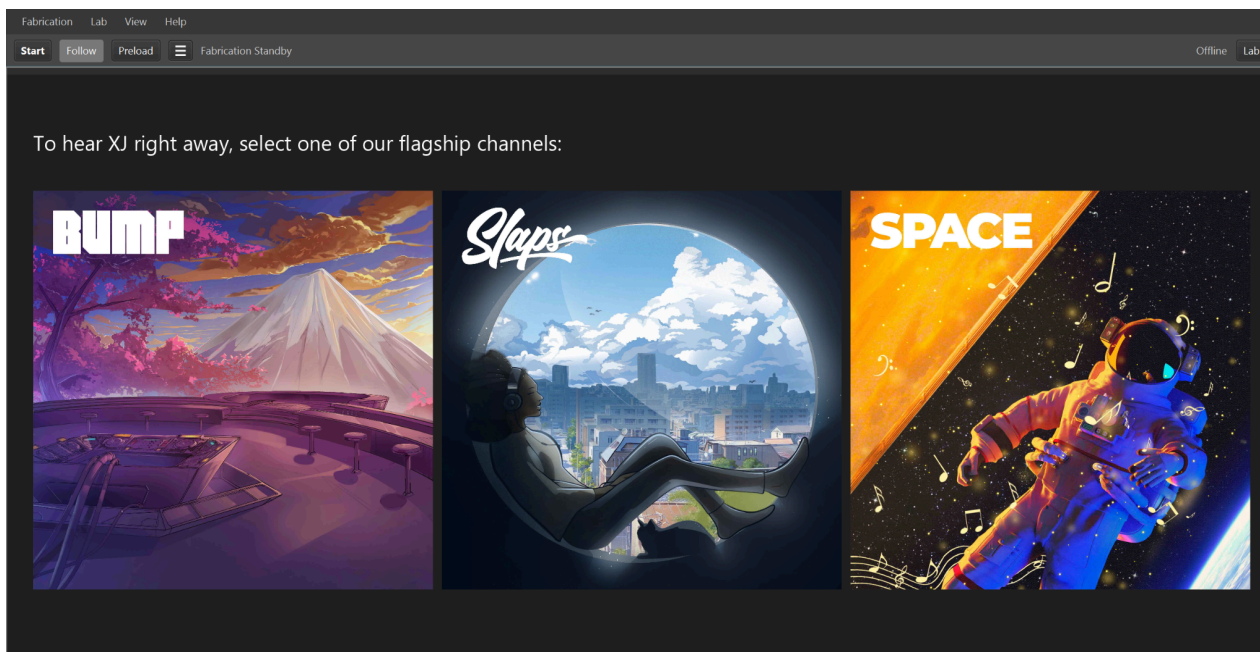
- Download the XJ music workstation Mac OS installer ZIP file to your chosen directory.
- Extract the contents of the ZIP file with your extraction tool of choice.
- Double-click the DMG file to mount the installer disk image as a volume.
- Open the XJ music workstation installer volume that appears your desktop (unfortunately, it is not named "installer" nor does it have an installer icon appearance)
- Inside the installer volume, drag the XJ music workstation application into your Applications folder.
- After installation, XJ music workstation will appear in the Applications folder as well as in the Spotlight Search results.
- The first time you try to start a new installation of XJ music workstation, Mac OS will refuse to open the application because this app is not signed. Choose the "View in Finder" option.
- In Finder, inside the Applications folder, right-click (option-click) on the XJ music workstation application and choose Open- this will override security and run the application.
- Repeat the previous step. For some reason, Mac OS forces this.

## Linux Installation

- Debian/Ubuntu users simply install the .deb package to make XJ music workstation available from the launcher.

# Getting Started in the Workstation

When first opening the XJ music workstation, the user will have the option of demoing three of the flagship music channels from our in-house artists: Bump, Slaps., and Space. These demos offer users a firsthand experience of XJ music's capabilities as a platform and can inspire creativity in their own sessions.



To begin playback on any of the three demo channels, the user will simply click on the corresponding artwork. While listening the user can preview the timeline and get a feel for the information displayed within it. By default, the timeline is set to Follow and will continue to scroll horizontally with the current playback position. To stop playback click the green Stop button at the top left corner above the timeline. To select a different channel, access the Fabrication settings or preload audio, click the same button again, which will now Reset the workstation.

## Fabrication Dropdown

The Fabrication Menu is the command center of the XJ music workstation, offering precise control over your compositions. The table below will walk you through each feature, ensuring you make the most of the creative possibilities at your fingertips.



Fabrication Settings - XJ music workstation ✕

Input Template ID:  Input Mode:

Output Mode:  Output File Mode:  Output File Seconds:  Output File Path Prefix:

Output Frame Rate:  Output Channels:  Craft Ahead Seconds:  Dub Ahead Seconds:  Ship Ahead Seconds:

Content Storage Path Prefix:  Timeline Segment View Limit:

Fabrication Menu Settings	Description
<b>Input Template ID</b>	Input the specific template ID to instruct the workstation on which template to play.
<b>Output Mode</b>	Playback Mode plays the selected Template back in real time. File Mode generates a recording of the Template faster than real time, but without playback.
<b>Output File Mode</b>	Controls how outputs in File Mode are divided. Continuous outputs one file, while Segment, Macro, and Main output files based on the lengths of each of the three options.
<b>Output File Seconds</b>	Sets the length of a file generated using the Continuous file mode.
<b>Output File Path Prefix</b>	The location where files will be saved.
<b>Output Frame Rate</b>	The frame rate at which sounds associated with the Template will be used or converted to before playback or output.
<b>Output Channels</b>	The number of channels utilized in playback and output.
<b>Craft Ahead Seconds</b>	The amount of time XJ will Craft ahead in seconds.

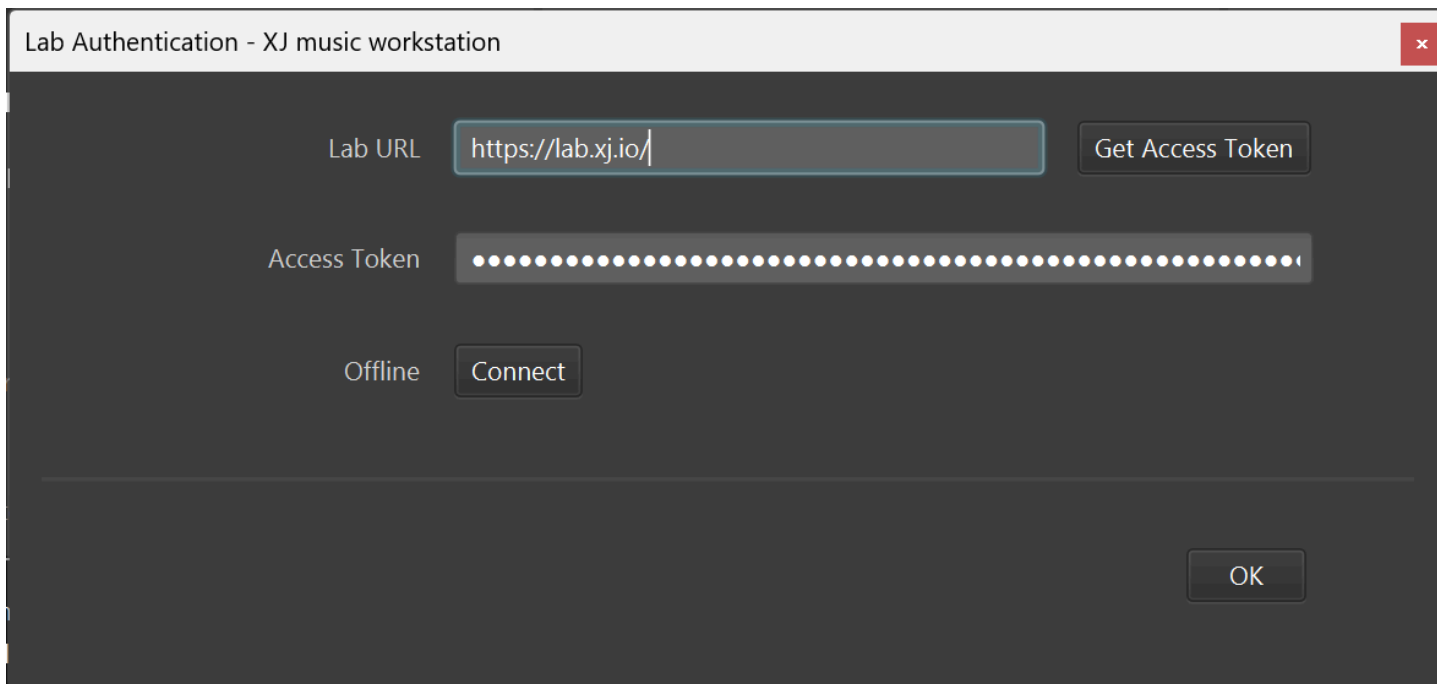
<b>Dub Ahead Seconds</b>	The amount of time XJ will Dub ahead in seconds.
<b>Ship Ahead Seconds</b>	The amount of time XJ will Ship ahead in seconds.
<b>Content Storage Path Prefix</b>	Sets the location where sound files associated with the chosen Template will be saved and preloaded from.
<b>Timeline Segment View Limit</b>	Limits segments available to view in the timeline to this value.

## Lab Dropdown

Lab Dropdown	Description
<b>Authentication</b> ( <i>Ctrl+L</i> )	Opens the Lab Authentication Menu.
<b>Open in Browser</b> ( <i>Ctrl+Alt+L</i> )	Opens the Lab in the user's preferred browser

## Lab Authentication Menu

The authentication menu is where users will have access to the Lab to create and access content within XJ.



Lab Menu	Description
<b>Authentication</b> ( <i>Ctrl+L</i> )	Opens the Authentication Menu.
<b>Lab URL</b>	Provides a URL address to the XJ Lab.
<b>Get Access Token</b>	Opens the preferences page of the Lab in your preferred browser, users will need to sign in using a Google account.
<b>Access Token</b>	Access tokens from the Lab are pasted here to establish a link between the XJ workstation and Lab.
<b>Connect</b>	Displays the current status of the connection between the XJ workstation and Lab, users will be Offline or Authenticated. When authenticated the avatar, account name, and email address of the user will be shown here.
<b>Disconnect</b>	Disconnects the user from the Lab.

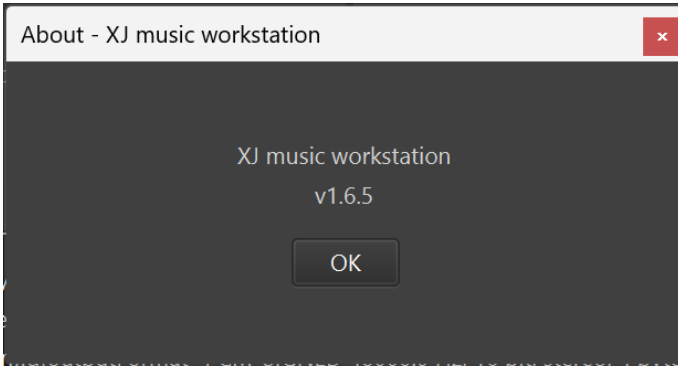
## View Dropdown

The View dropdown contains options for changing the workstations appearance and viewing Logs.

View Menu	Description
<b>Theme</b>	Allows the user to switch between Dark Theme and Light Theme
<b>Show Logs (F2)</b>	Opens the Log window at the bottom of the Workstation, allowing the user to view the status and progress of ongoing processes.
<b>Logs Auto-scroll (Ctrl+F2)</b>	Access tokens from the Lab are pasted here to establish a link between the XJ workstation and Lab.
<b>Log Level</b>	Allows the user to select which levels of information they will be shown in the logs. There are options to toggle Debug, Info, Warning, and Error logs on and off.

## Help Dropdown

Help Menu	Description
<b>User Guide</b>	Opens up this User Guide.
<b>About</b>	Displays the current version of the XJ music workstation.



## Creating in the Lab

### Sign in

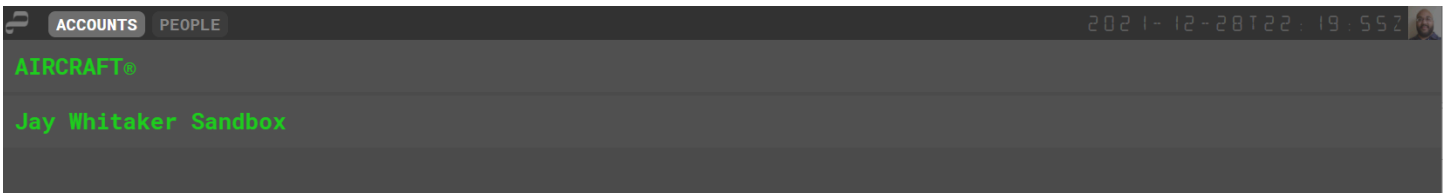
Artists sign in to <https://lab.xj.io> with Google



One person logging in to collaborate on XJ content is called a User. Users are personalized profiles for each artist, allowing them to work in the XJ Lab, behind a set of authentication measures.

In XJ, an Account is for an Organization to silo their content for rights management purposes. For example, the band AIRCRAFT has their own account.

At any time, click on the logo in the upper left corner to see the list of all accounts available to you:

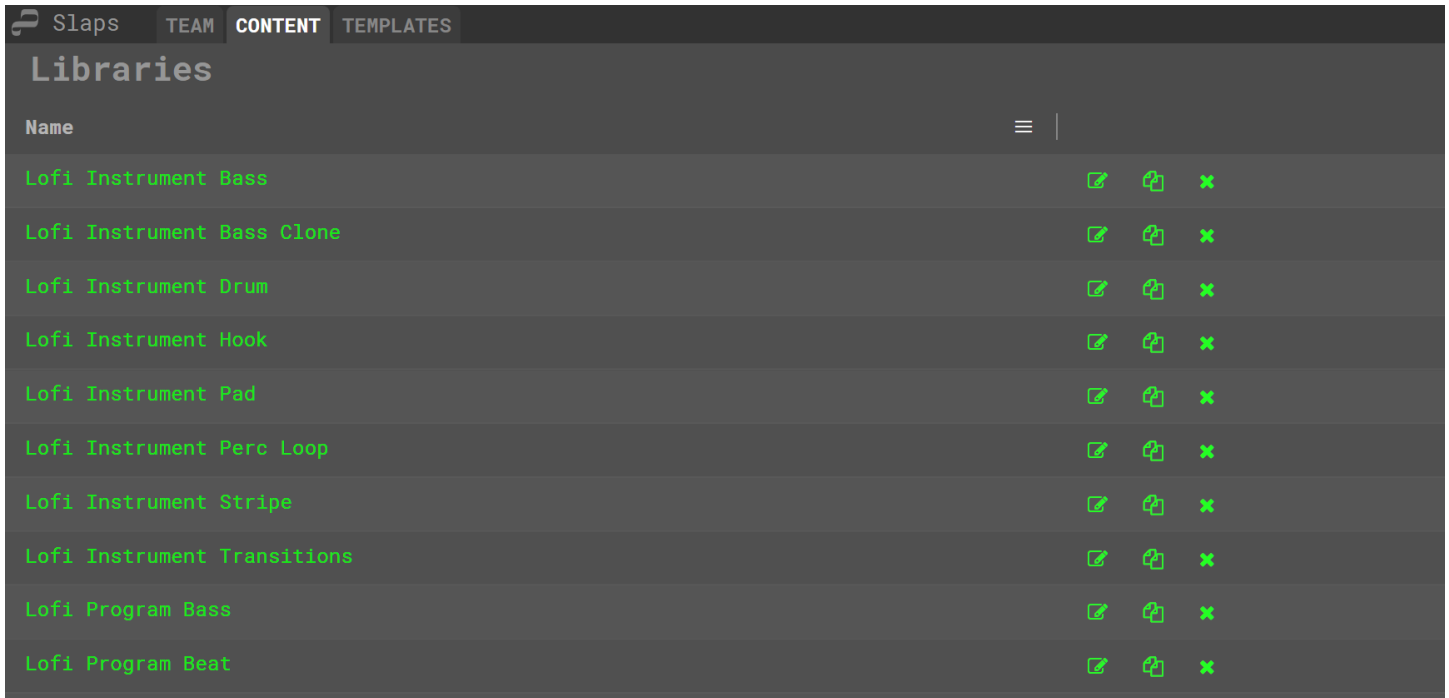
































Click on the name of an Account to see all the content within.

# Libraries

Libraries are collections of content within the XJ Lab.

When you first open an account, you'll see a list of all its Libraries.



Name			
Lofi Instrument Bass			
Lofi Instrument Bass Clone			
Lofi Instrument Drum			
Lofi Instrument Hook			
Lofi Instrument Pad			
Lofi Instrument Perc Loop			
Lofi Instrument Stripe			
Lofi Instrument Transitions			
Lofi Program Bass			
Lofi Program Beat			

Click on the name of a Library to see the content within.

Libraries contain two types of content: [Programs](#) and [Instruments](#).

## Pro Tip

The buttons on each row can be used to **Edit**, **Duplicate**, or **Destroy** that Library.



# Programs

In XJ music, a **Program** is a collection of musical instructions. While all programs are stored according to the [Program Model](#), different [Types](#) of programs have unique purposes and behaviors.

Each program contains a [configuration](#), which may affect the music when that program is chosen.

The program menu contains a list of all programs.

Here you can see the attributes of each Program, as well as the memes assigned to it.

Type	State	Name	Key	Density	Tempo
Beat	Published	<a href="#">Don't Have a Cow Man</a>	X	1	85
Beat	Published	<a href="#">Lo Rider</a>	C	1	75
Beat	Published	<a href="#">Lo Rider 2</a>	C	1	75
Beat	Published	<a href="#">Lo Rider 3</a>	C	1	75
Beat	Published	<a href="#">Lo and Behold</a>	C	1	75
Beat	Published	<a href="#">Lo-Falutin</a>	X	1	85
Beat	Published	<a href="#">Slo-Falutin</a>	X	1	75
Beat	Published	<a href="#">Sweet and Lo</a>	C	1	75

To compose the content of any given program, click its name in the list to open the [Program Editor](#).

**Pro Tip**  
The buttons on each row can be used to **Move, Edit, Duplicate, or Destroy** that Program.

## Types of Programs

Type	Purpose
Macro Programs	Like a playlist, this describes an order in which to play songs a.k.a. main programs.
Main Programs	Like a song, this musical notation describes chord progressions, as well as voicings of those chords for various types of instruments.
Beat Programs	This describes a beat, as in the foundation of a groove. XJ chooses one beat for an entire song (Main Program). This musical notation is used directly for a Drum Instrument to play.
Detail Programs	This is writing down the rhythm of a part for one instrument to play, in a way that may be interpreted by XJ in for any song according to its chord voicings.



# Configuration of Programs

Details of program configuration

# Instruments

In XJ music, an **Instrument** is a collection of Audios. While all instruments are stored according to the [Instrument Model](#), different [Types](#) and [Modes](#) of instruments have unique purposes and behaviors.

Each instrument contains a [configuration](#), which may affect the music when that instrument is chosen.

The instrument menu contains a list of all instruments.

Here you can see the attributes of each Instrument, as well as the memes assigned to it.

Type	Mode	State	Name	Volume	Density
Bass	Event	Draft	Chill Bass 01	1.6	0.6
Bass	Event	Published	Chill Bass 01 V2	2.5	0.6
Bass	Event	Draft	Chilled Water Long Bass 01	2.2	0.6
Bass	Event	Draft	Earth Large Bass	1	0.6
Bass	Event	Published	Earth Large Bass 2 (Pbass)	1.2	0.6
Bass	Event	Draft	Earth Small Bass	1	0.6
Bass	Event	Published	Earth Small Bass 2	0.9	0.6
Bass	Event	Draft	Water Large Bass	1	0.6

To compose the content of any given instrument, click its name in the list to open the [Instrument Editor](#).

**Pro Tip**  
The buttons on each row can be used to **Move, Edit, Duplicate, or Destroy** that Instrument.

## Types of Instruments

Type	Purpose
Drum	For use in a beat, e.g. KICK or SNARE
Bass	Paired with detail program (event mode) or standalone (chord mode)
Pad	Paired with detail program (event mode) or standalone (chord mode)
Sticky	Paired with detail program (event mode) or standalone (chord mode)
Stripe	Paired with detail program (event mode) or standalone (chord mode)
Stab	Paired with detail program (event mode) or standalone (chord mode)
Hook	Used in loop mode, one selected per program
Percussion	<a href="#">Percussion loops</a> are layered and repeated behind the main beat. Percussion can be used with transition mode too.
Sweep	Used with Transition or Background
Noise	Used with Transition or Background

## Modes of Instruments

Type	Use a Detail Program?	Audio playback
Event	Yes	Multiple note-based audios to fulfill each multi-note event of detail program
Chord	No	One chord-based audio to fulfill each main program chord
Loop	No	Loop continuously; audio chosen for main program voicing at start of loop.
Transition	No	Layered over the seams between songs and sections
Background	No	Textural sonic wallpaper laid behind the music
Performance	No	Bound directly to Main Program Sequence (over specific chord changes)

## Instrument Type-Modes

	Event	Chord	Loop	Transition	Background	Performance
Drum	Green	Grey	Grey	Grey	Grey	Green
Bass	Green	Grey	Grey	Grey	Grey	Green
Pad	Green	Green	Grey	Grey	Grey	Green
Sticky	Green	Green	Grey	Grey	Grey	Green
Stripe	Green	Green	Grey	Grey	Grey	Green
Stab	Green	Green	Grey	Grey	Grey	Green
Hook	Grey	Grey	Green	Grey	Grey	Green
Percussion	Grey	Grey	Green	Green	Green	Grey
Sweep	Grey	Grey	Grey	Green	Green	Grey
Noise	Grey	Grey	Grey	Green	Green	Grey

### Hook-type Instruments (Loop-mode only)

A hook type instrument audio has a tones value storing notes that have been played during the audio recording, e.g. C4, D4, F4, A#5

Main programs have Hook-type voicings.

Future: [Voicings can be associated with Sequence \(not just one chord\)](#)

For example, here's a chart of melodic voicings for the main programs of Cool Air:

[Main Program Melodic Spreadsheet | Cool Air](#)

### Event-mode Instruments

A polyphonic instrument such as a Bass-type instrument will provide a single audio per note. In that case each row (audio) in the instrument will use the *tones* column to store the note. The *event* column is ignored.

A Drum-type instrument will provide a single audio per drum event. In that case each row (audio) in the instrument will use the *event* column to store the event. The *note* column is ignored.

### Chord-mode Instruments

A chord-mode instrument such as a Pad-type instrument will provide a single audio per chord. In that case each row (audio) in the instrument will use the *tones* column to store the chord. The *event* column is ignored.

When choosing instrument audio (from a chord-mode instrument) for the chords of a main program, XJ will prioritize (see: [marble bag](#)) any available exact match. In lieu of an exact match, XJ will allow an acceptable substitute of the pre-slash description of a chord to a non-slash chord (e.g. "Eb/G" is an acceptable substitute for "Eb" and vice versa).

While the presence of voicings is used to make the decision "I'm going to look for a \_\_\_-type instrument" once XJ has arrived at the decision "I'm going to craft this \_\_\_-type chord-mode instrument now for this segment" there are zero references to voicing content. The content of the chord voicings is ignored. E.g. It's possible to tell XJ to seek Pad instruments for a given main program even if the content of those voicings are all just "X" but it's critical to understand the whole template you are composing. If the design of the template calls for both Note- and Chord-type instruments for a given instrument type, then it's necessary to write whole voicings (comma-separated lists of notes voicing each chord) for interoperability.

For a table of XJ's chord comprehension, see [Appendix A. Chord Interpretation](#)

## Performance-mode Instrument

[Artist plays along to stream to contribute new performances to the library](#)

### [Performance-mode Instrument](#)

Artist records and uploads performance-mode instrument wherein each audio has a `tones` value equal to the encoded chord changes over which it is performance

Any series of chord (e.g. a main program sequence) can be reduced to a string, a finite series of characters, e.g. 4\_C\_4\_Dm means 4 beats of C, followed by 4 beats of D minor.

Chord series are nestable, meaning that if a main program calls for 4\_C\_4\_Dm\_4\_F\_4\_Gm then XJ may choose any combination of pieces that fulfill it.

Needs to have safeguards to prevent excessive business, and template configuration for the density of this.

Fabrication chooses a performance-mode instrument and uses it over compatible chords, with no DP

## Configuration of Instruments

Details of instrument configuration

### Instrument Audio Quantization

When enabled, specify grain Q. Whenever an audio from this instrument is picked, the start transient of the audio will be aligned with the nearest beat rounded to the nearest Q beats- then the in and out point of that audio pick will be adjusted to the target in and out time.

E.g. for  $Q = 1.0$ , pick audio A at segment beat 1.5, the result is that this audio is heard starting at beat 1.5, but played from 0.5 beats in to the audio, such that the playback of audio aligns as if it had been started on beat 1.0.

```
instrumentAudioQuantization=true
```

[Quantization of instrument audio and detail programs](#)

# Templates

Templates are packaged compilations of Libraries, potentially highlighting specific Instruments and Programs that can be previewed and/or broadcast.

Each template contains a [configuration](#), which may affect the music when fabricating that template.

Name	Type
Mark Stewart's Preview Template	
Slaps: Lofi	
Slaps: Lofi Dev	
Slaps: Lofi Multiphonic Test	
Slaps: Trap Dev	
Slaps: Voodoo Lion Dev	

Click on the name of a Template to edit that template.

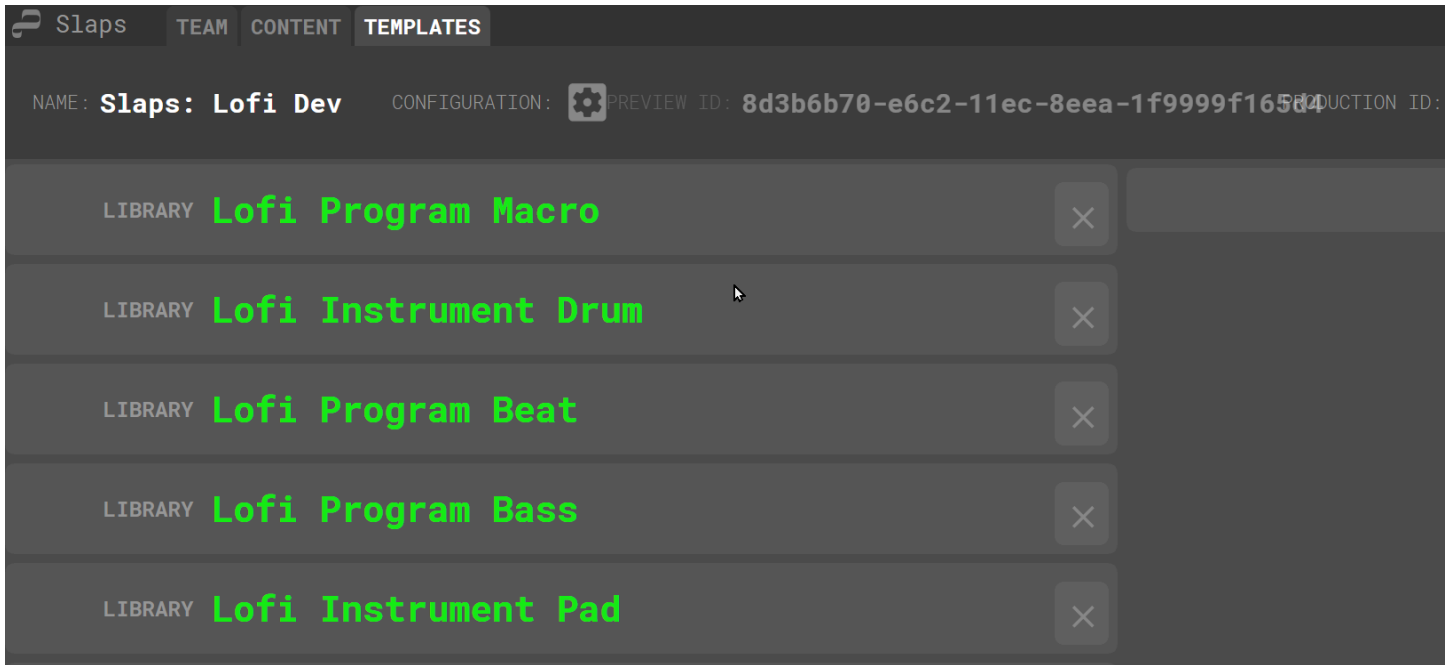
### Pro Tip

The buttons on each row can be used to **Edit**, **Duplicate**, or **Destroy** that Template.



## Binding Template Content

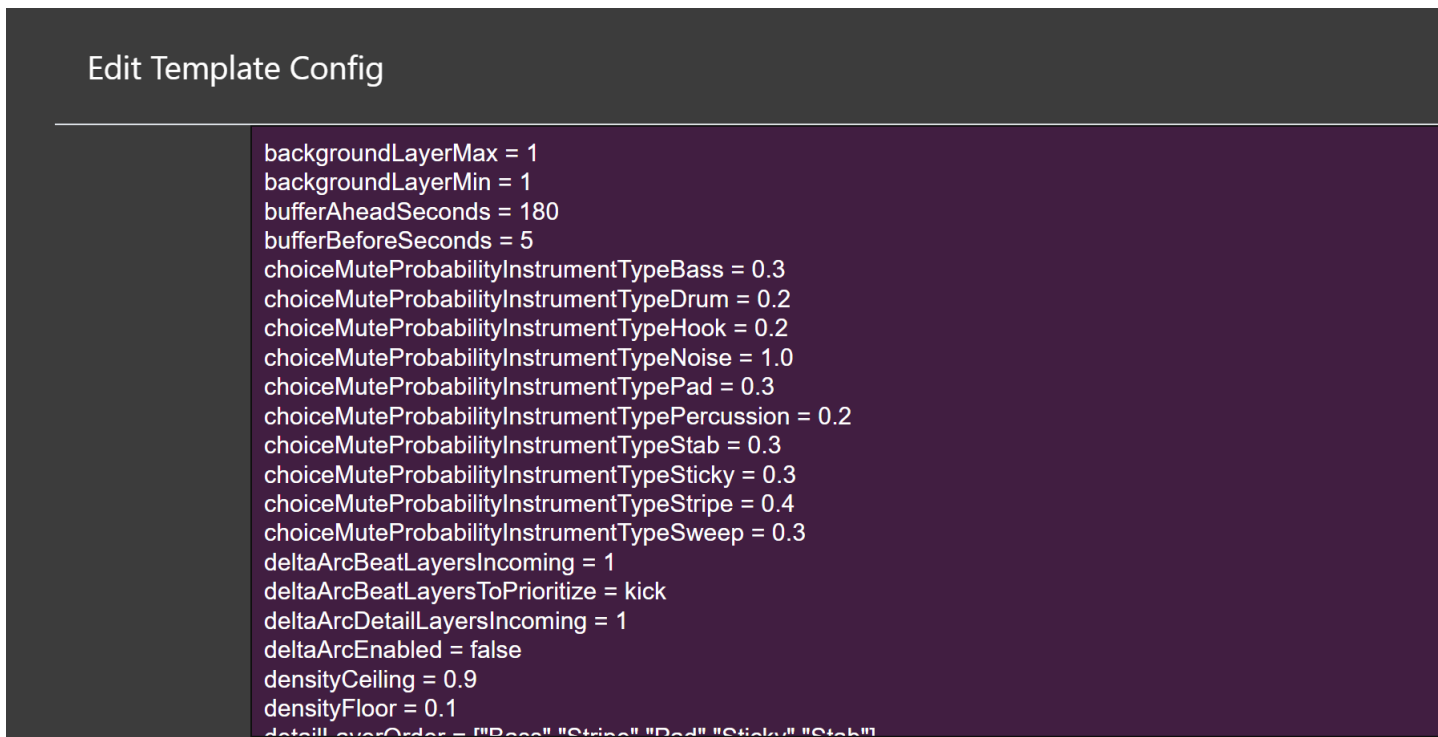
The Bindings menu allows you to specify which Library, Program, or Instrument can be utilized in a Template. When creating a Template it is important to Bind all of the necessary Libraries, including Main and Macro Programs, Beat Programs, Detail Programs, and Instruments.





## Configuration of Templates

Each Template contains a Template Configuration, a set of values and definitions artists provide to XJ in addition to their chosen Bindings.



The Template Configuration menu is accessed by clicking the gear icon in the middle of the Template header.

Template Configuration Values	Description
backgroundLayerMax backgroundLayerMin	The maximum and minimum number of background Ambience Loop events selected
bufferAheadSeconds bufferBeforeSeconds	The amount of seconds that the Template will buffer ahead and before*
choiceMuteProbabilityInstrumentType	The chance that the Instrument type will be randomly muted in a sequence, from 0.0-1.0 (0% to 100%)
deltaArcBeatLayersIncoming	The number of Beat Layers being input into the Delta Arc function
deltaArcBeatLayersToPrioritize	The Events in the Sequence to be prioritized in the Delta Arc function
deltaArcDetailLayersIncoming	The number of Detail Layers being input into the Delta Arc

	function
deltaArcEnabled	Enables or disables the Delta Arc function
densityCeiling densityFloor	The upper and lower limits of Sequence Density (0.0-1.0)
detailLayerOrder	The order in which Instrument types are prioritized
dubMasterVolumeInstrumentType	The master volume of an Instrument type
eventNamesLarge eventNamesMedium eventNamesSmall	Defines the naming convention for Event sizes, allowing for similar terms like “small”, “little”, or “low” to all be recognized as Small Events, for example
instrumentTypesForAudioLengthFinalization	Determines which Instrument types are muted when another instance of itself plays or the sequence ends
instrumentTypesForInversionSeeking	Determines which Instrument types are able to have their chords inverted
mainProgramLengthMaxDelta	Sets the length of Main Program as they relate to Delta Arcs
memeTaxonomy	The words or phrases XJ will define as Memes (Meme) and the categories that Memes fall into (Category)
mixerCompressAheadSeconds mixerCompressDelaySeconds	
mixerCompressRatioMax mixerCompressRatioMin	
mixerCompressToAmplitude	
mixerDspBufferSize	
mixerHighpassThresholdHz mixerLowpassThresholdHz	The upper and lower thresholds for the High Pass and Low Pass EQs.
mixerNormalizationBoostThreshold mixerNormalizationCeiling	
outputChannels	The number of output channels
outputContainer	Defines the the output container used, e.g ‘OGG’
outputContentType	Defines the
outputEncoding	
outputEncodingQuality	

outputFrameRate	The output frame rate
outputSampleBits	The output bit-depth
percLoopLayerMax percLoopLayerMin	The maximum or minimum amount of Percussion Loop Layers that will play in the Chain
stickyBunEnabled	Enables or disables the Sticky Buns function
transitionLayerMax transitionLayerMin	The maximum or minimum amount of Transition Layers that will play in the Chain
vmResourceLimitCpu	* it isn't recommended that artists make changes to this setting.
vmResourceLimitMemoryGb	* it isn't recommended that artists make changes to this setting.
vmResourceRequestCpu	* it isn't recommended that artists make changes to this setting.
vmResourceRequestMemoryGb	* it isn't recommended that artists make changes to this setting.

As new features are rolled out to XJ, it is important to periodically update your Template Configurations by scrolling to the bottom, hitting enter, and saving. When you open your Configuration again, new settings might appear and then be able to take effect in your Preview Template.

## Ship Keys

Chains are shipped with unique identifiers called keys. When troubleshooting a chain, it is important to try inputting a new ship key to manually reset the chain.

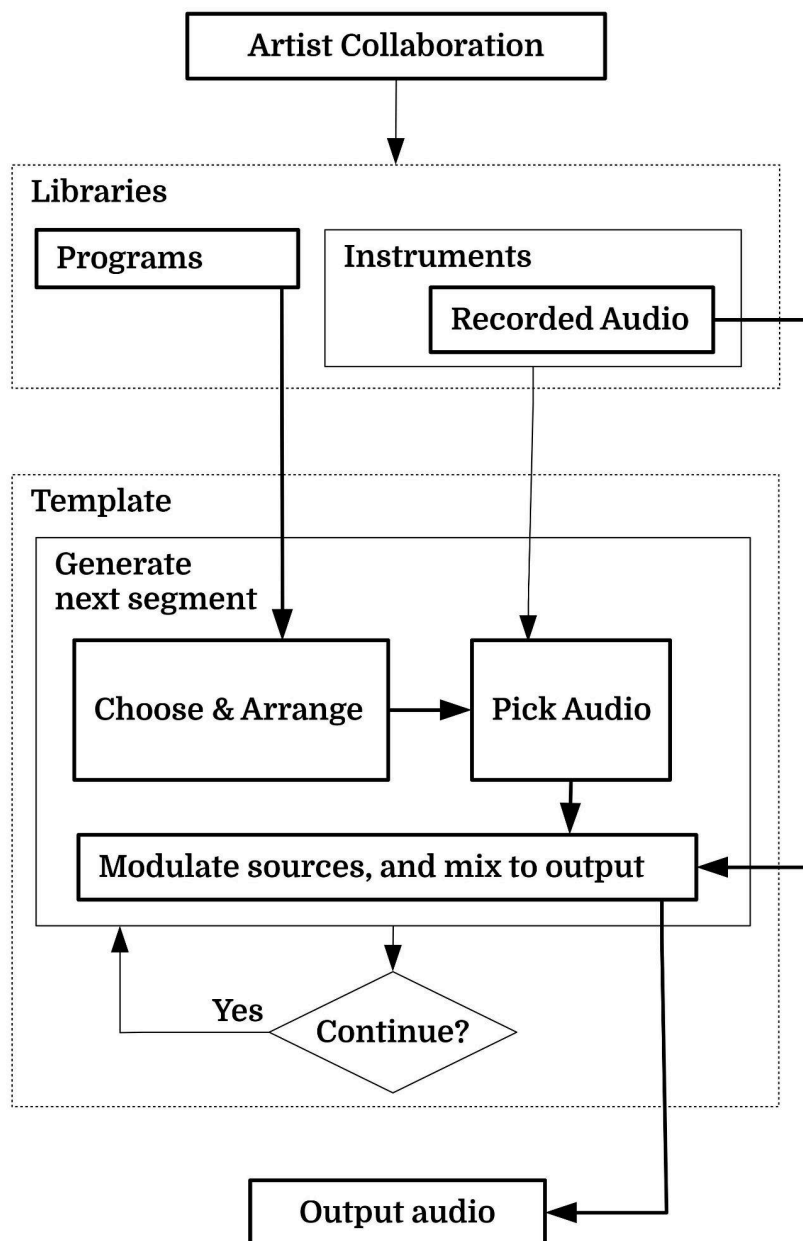
## Fabrication of Segments

XJ is essentially a traditional sampler-sequencer,<sup>1</sup> extended with new generative features.

Artists create a **Template** (Libraries of **Programs**, **Instruments**, and their **Recorded Audio**).

XJ fabricates a continuous series of audio **Segments**.

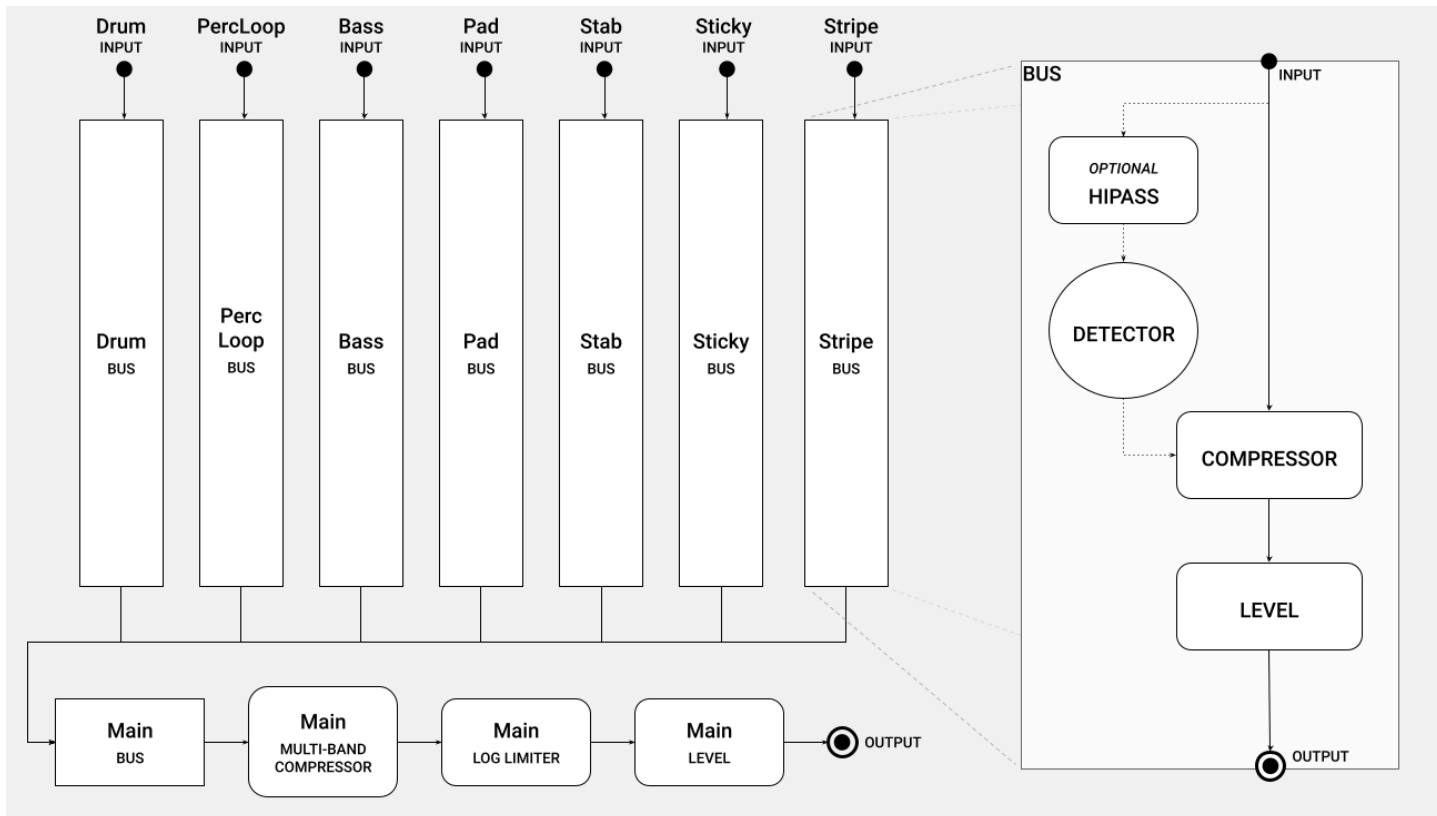
For each Segment, XJ chooses and arranges instrument audio, then modulates sources and mixes **output audio**.



<sup>1</sup> [Sampler \(musical instrument\) - Wikipedia](#)

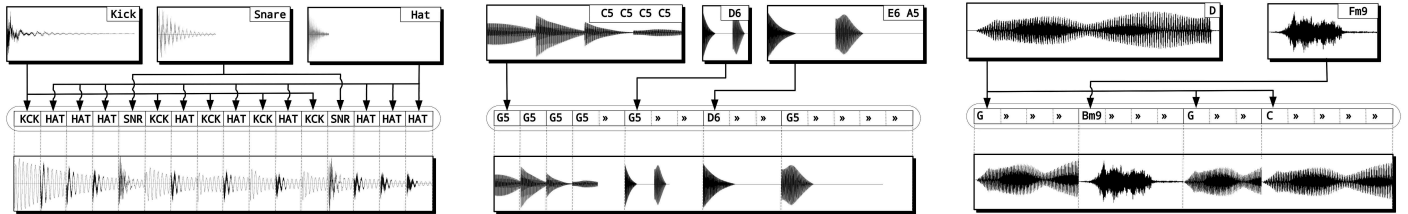
## Dubbing of Output Audio

We are still working with the original simple XJ audio mixer, to combine all source audio into the final output audio. Following is the design for our v0.7 Mixer, as implemented by the Pivotal story [Mixer bus/compressor enhancement \(#178919097\)](#)

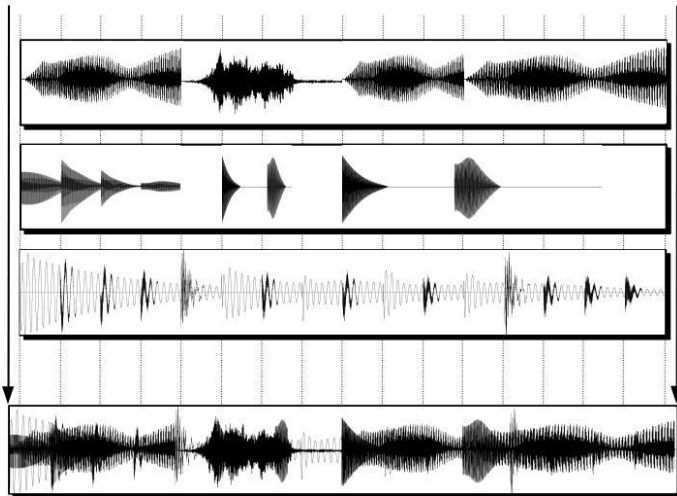


Bring in the diagrams from the patent in order to illustrate how audio makes its way from instruments, through choice, arrangement, and picking, then finally to dub

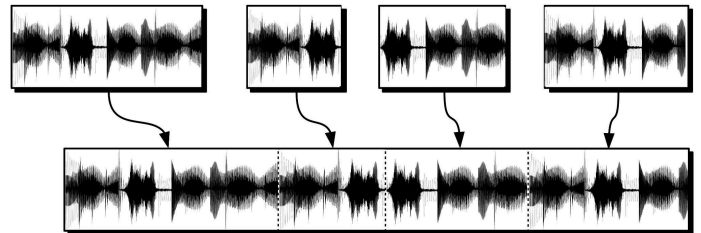
### 1. Mix each Layer from arranged Instrument Audio sources



### 2. Mix Layers into Segment



### 3. Add Segments for Streaming



## Program Editor

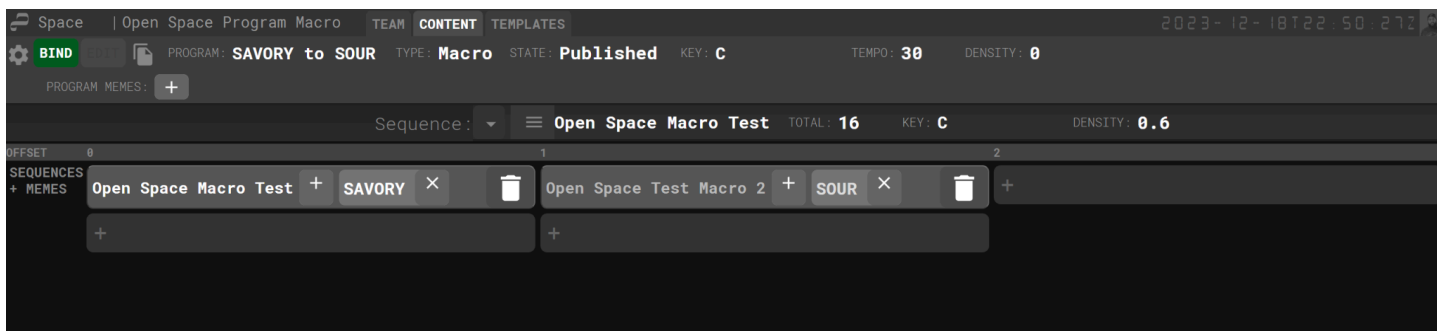
The Program Editor allows you to edit music notation by placing Events within Patterns.

There are different modes of editing available. The editor will automatically show the panels corresponding to the type of program you are currently editing: [Macro Program](#), [Main Program](#), [Beat Program](#), or [Detail Program](#).

The purpose of each of these four types of programs is described in [this table](#).

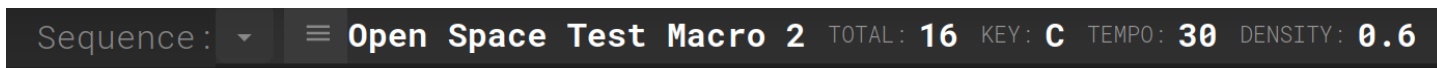
## Editing a Macro Program

Each sequence of a macro program corresponds to a song a.k.a. main program that will be chosen.

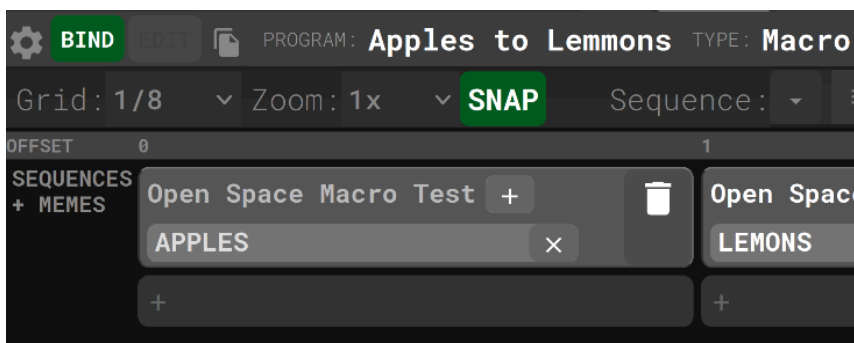


The content of a macro program sequence is irrelevant. We only use a sequence to store attributes such as tempo and density. For example, this is how tempo is driven. However, we are waiting on some software enhancements such as [XJ Tempo Behavior v2](#) and a [more user-friendly editor mode for macro programs](#).

First, you'll create some number of sequences containing attribute values for total, key, tempo, and density:



Next, you'll make sure to be in BIND mode (see the upper left corner) and then bind your sequences into some order. The bindings can have unique memes. **This whole process is a real pain that needs to be simplified.**



But for example, in Slaps we are using it to regulate between three tempos. The tempos are also correlated with memes, but the fact that these memes actually correspond to changes in tempo is dictated by the macro programs.



# Editing a Main Program

## Chord search feature

Space | Open Space Program Main TEAM **CONTENT** TEMPLATES

⚙️ BIND **EDIT** PROGRAM: **Open Space Csus2 check2** TYPE: **Main** STATE: **Published** KEY: **C** TEMPO: **30** DENSITY: **0**

PROGRAM MEMES: + **CMAJ** X **SPICY** X

Grid: 1/8 Zoom: 1x **SNAP** Sequence: A-7add9 TOTAL: 64 KEY: C DENSITY: 0

Sequence Chords + A-7add9

VOICE: **Bass**

A1, E2, A2, C3, E3, G3, A3, C4

# Editing a Beat Program

The screenshot shows the 'EDIT' view of a beat program. At the top, the program name is 'Don't Have a Cow Man' with a type of 'Beat' and a state of 'Published'. The tempo is set to 75BPM and there is a 'BEEF' meme. The interface is divided into two main sections: 'Kick Snare Hats' and 'HIHATCL'. The 'Kick Snare Hats' section shows a 'Drum' voice with 8 total items. The 'HIHATCL' section shows a 'KICK' voice with 1 total item. The 'KICK' voice has a value of 1 and an 'X' mark. The 'HIHATCL' voice has values of 0.02, 0.49, and 0.9, with an 'X' mark. The 'Main' sequence is selected, and the total number of items is 16.

Slaps | Lofi Program Beat | TEAM | **CONTENT** | TEMPLATES

PROGRAM: **Don't Have a Cow Man** TYPE: **Beat** STATE: **Published** KEY: **X** TEMPO: 75BPM

PROGRAM MEMES: + 75BPM X BEEF X

Grid: 1/8 Zoom: 1x **SNAP** Sequence: Main TOTAL: 16 KEY:

**Kick Snare Hats**  
VOICE: **Drum**  
1  
TOTAL: 8

**HIHATCL**  
0.02  
0.49  
0.9  
X

**KICK**  
0  
0.44  
1  
X

## Editing a Detail Program



The editor will appear in one enter will be a mode Depending on the type of program you are editing,

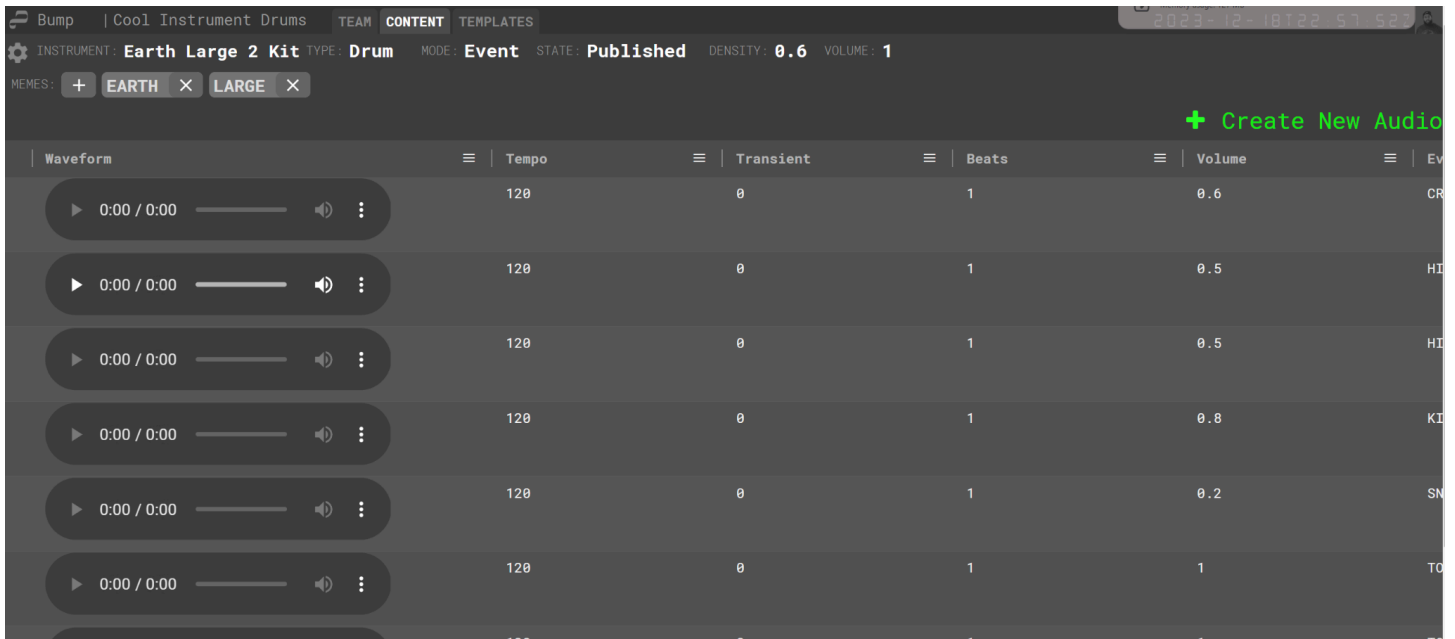
Event length and velocity is controlled by clicking and dragging the vertical and horizontal edges of each event, respectively.

(coming soon)

## Instrument Editor

In XJ music, an instrument comprises a set of **Instrument Audios**.

Also, the Instrument has some attributes shown at the top of the editor



Pending the completion of [Lab UI instrument audio editor: set audio transient seconds by clicking on waveform](#) we will add screenshots of the waveform editor and its use

### Pro Tip

The buttons on each row can be used to **Download, Upload, Move, Edit, Duplicate, or Destroy** that Instrument Audio.

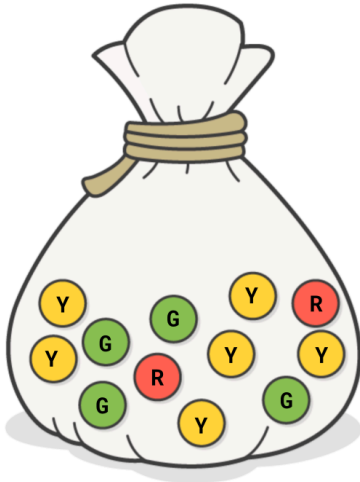


# Making XJ music

## Memes Influence Choice

A meme is a tag that artists apply to programs and instruments in order to guide the choices that XJ makes.

XJ factors in the memes it sees in order to make **choices** about which programs and instruments will be included in a Segment.



### Pick a Marble from the Bag

In order to make random choices based on weighted options, XJ uses a stochastic<sup>2</sup> mechanism called a Marble Bag.

The example figure to the left represents a choice between three possible outcomes: Red (R), Green (G), and Yellow (Y). The likelihood of a choice is increased when we add more of that choice's marbles to the bag.

In the following section, we'll discuss how the presence of different kinds of **memes** either increases or decreases the chance of a selection, or prevents a particular option from being chosen at all. These are in fact expressed by adding and removing marbles from a virtual bag.

For example, to make sure an option is never chosen, we remove all of its marbles from the bag.

detail the process of how we add memes to the segment as we choose content

XJ's marble bag is actually divided into phases. When a marble is put into the bag, it is assigned a phase. For example, if the phase 1 bag contains any marbles, we will pick from only the phase 1 bag and skip phases 2 and beyond. This supports functionality such as "XJ always chooses a directly-bound program or instrument when available"

---

<sup>2</sup> <https://en.wikipedia.org/wiki/Stochastic>

## Types of Memes

Notes about Meme Stack being Axioms ... Theorems

Called	Example	When meme matches
Regular Meme	<b>PEACHES</b>	Increase likelihood to choose content that matches regular memes XJ has already chosen for the segment.
Anti-Meme <i>aka. Not-Meme</i>	<b>!PEACHES</b>	Do not choose this content, if we have already chosen content with the corresponding regular meme.  If XJ has already chosen content for the segment having anti-meme <b>!PEACHES</b> , then XJ will not choose content having regular meme <b>PEACHES</b>
Unique-Meme <i>aka. Solo-Meme</i>	<b>\$PEACHES</b>	Do not choose this content, if we have already chosen content with this unique-meme.  If XJ has already chosen content for the segment having unique-meme <b>\$PEACHES</b> , then XJ will not choose any other content having this same unique-meme.
Number-Meme	<b>5PEACHES</b>	Do not choose this content, if we have already chosen content with a different number of the same meme.  If XJ has already chosen content for the segment having number-meme <b>5PEACHES</b> , then XJ will not choose content with number-meme <b>7PEACHES</b> .
Strong-Meme	<b>PEACHES!</b>	Only choose content if we have already chosen content with the corresponding regular meme.  Only if XJ has already chosen content for the segment having regular meme <b>PEACHES</b> will XJ consider choosing content with strong meme <b>PEACHES!</b>  * Strong memes are <a href="#">not added to the stack</a> .

## Meme Taxonomy

A template configuration has a field called **memeTaxonomy** which defines the taxonomy of memes.

For example, this might look like

```
memeTaxonomy = [  
  {"memes": ["RED", "GREEN", "BLUE"], "name": "COLOR"},  
  {"memes": ["WINTER", "SPRING", "SUMMER", "FALL"], "name": "SEASON"}  
]
```

That would tell XJ about the existence of a meme category called **COLOR** with values **RED**, **GREEN**, and **BLUE**, and a meme category called **SEASON** with values **WINTER**, **SPRING**, **SUMMER**, and **FALL**.

For example, after content having **RED** is chosen, we can choose nothing with **GREEN** or **BLUE**.

*NOTE: Use a semicolon to separate the categories. Each category has a comma-separated list of memes.*

## Instrument Behavior

Different [Types](#) and [Modes](#) of instruments have unique purposes and behaviors.

Write up the definitive guide to how XJ handles various combinations of instrument types and modes

*Option A, is to tag audios with a single chord symbol (ie Cmaj7, Emin7b5) which is itself a shorthand for a list of notes (C, E, G, B and E, G, Bb, D for these examples)*

*With some clever voicing list tricks you could have a lot of freedom with this route. For example if you wrote a voicing list that included both Fmaj7 and Amin7 you would be outlining Fmaj7/9*

*The more complicated and versatile route, Option B, is what you're saying—tagging audios with comma-separated lists of notes and having XJ cross-reference those with Main Program voicing lists to choose audios.*

*I think both will be useful, but to me the chord symbol tagging seems like a faster route to get multi-note audio happening because it's the same process as the current single-note tagging. I could be underestimating what's going on currently though*

*Theoretically, option B is closer to what exists today, but that's not an obstacle.*

*I believe that there is a simple formula to tell us whether Option A or Option B yields more freedom a.k.a. modularity a.k.a. less work for more payoff.*

*Starting with Template Analysis (soon to be built) we can see how many different chords are present in all the main programs of the given template*

*(For each type of instrument independently?) Calculate the relative complexity of using Chord-based (option A) or Note-based (option B) audio*

*Chord-based complexity = How many different audios would be required to cover all the chords present in all main programs*

*Note-based complexity = How many different audios would be required to cover all the voicing lists present in all main programs*

*I suspect the answer (A vs B) varies between templates and instrument types*



## Events

There are two types of **Events**, the **Sequence Pattern Voice-Event** and the **Instrument Audio-Event**. Automatic fabrication performs a search for matching Instrument Audio-Events for the chosen Sequence Pattern Voice-Events. That is how an Audio is chosen inside of an Instrument, based on its Audio-Events.

Write up all details about how events from the programs lead to events in the audios which lead to mixing decisions. There are some interesting diagrams from the patent to share here as well.

Add some info about how we use a score-based stochastic model for event audio picking, as opposed to the marble bag model we use for choice.

## Drum Beat Events

The beat program is the center of the groove during any given main program. XJ chooses a single beat and keeps that beat for the duration of the main program.

It is important for percussive-type instruments to use the following standardized set of Event Names, which have far more influence on the outcome of percussive instrument-audio choice (than, for example, Event Notes).

Currently, we are using a very narrow set of events for in Beat-type programs.

We currently use these codes, for beat program events, and drum instrument audio events:

CODE	Name
HIHATCL	High Hat Closed
HIHATOP	High Hat Open
KICK	Kick
SNARE	Snare

## Drum Detail Events

In the future, XJ will also choose Detail programs having additional drum events to decorate the groove. These drum detail programs serve a similar function as [percussion loop](#) and transition instruments.

We expect that these events may include various types of drums or other percussion instruments.

We plan to use these codes for drum detail program events, and drum instrument audio events:

CODE	Name
BONGOHI	Bongo High
BONGOLO	Bongo Low
CONGAHIMU	Conga High Mute
CONGAHIOP	Conga High Open
CONGALO	Conga Low
TIMBALEHI	Timbale High
TIMBALELO	Timbale Low
TOMFLRHI	Tom Floor High
TOMFLRLO	Tom Floor Low
TOMHI	Tom High
TOMHIMID	Tom High Mid
TOMLO	Tom Low
TOMLOMID	Tom Low Mid

CODE	Name
BLOCKWDLO	Block Wood Low
CABASA	Cabasa
CLAP	Clap
CLAVES	Claves
COWBELL	Cowbell
CUICAMU	Cuica Mute
CUICAOP	Cuica Open
CRASH	Cymbal Crash
RIDE	Cymbal Ride
SPLASH	Cymbal Splash
GONG	Gong
GUIROLN	Guiro Long
GUIROSH	Guiro Short
MARACAS	Maracas
SHAKER	Shaker
SLAP	Slap
STICKSIDE	Stick Side
TAMBOUR	Tambourine
TRIANGMU	Triangle Mute
TRIANGOP	Triangle Open
VIBRASLAP	Vibraslap

## Percussion Loops

Percussion Loops have an [Instrument Type](#) of Percussion, and an [Instrument Mode](#) of Loop.

They are assigned for each fabricated segment during percussion loop craft. It's going to be used to determine a target # of perc loops.

Assumed that one or more loops will be chosen for any segment— minimum and maximum are configured via template config. For each MP, setup delta arcs for N number of percussive loops.

Perc loop instrument behavior selects one audio per section. Consolidating audio with the same sound set under one instrument. Then setting the default length an audio stays in the stream to a full section rather than switching every 4 beats.

Percussive Loops can specify # beats in the instrument / audio

## Loop Priority

Some Perc Loops are killer by themselves. In the case of house music, this might be a fully fleshed out 16x 16th notes of house locomotion, requiring only an added kick and snare to be complete. Let's call these `PRIMARY` perc loops.

Other Perc Loops are intended to be added on top of a stable base of Primary perc loops. Let's call these `SECONDARY` perc loops.

Other Perc. Loops are intended to support the known structure of the rhythm. Let's call these `BACKBEAT` perc loops.

We can store the priority in audio name, similar to how we're using name for transition loop  
small/medium/large

## Background Ambience

### [New Background-type Instrument](#)

audio is at least 1 minute of background tone (longer than any section)

XJ uses a random piece from within the available audio to fill each section when the instrument is chosen.

XJ builds in and out of layers of ambience the same way it does perc loops.

## **Arcs**

Describe delta & section arcs

Opportunity to change these from video game demo

# Appendix A. Chord Interpretation

Using 12-tone equal-tempered octaves.

Chord	Synonyms	Chord	Synonyms	Chord	Synonyms	Chord	Synonyms
<b>C</b>	C maj C major CM	<b>C -</b>	C min C minor Cm C mi	<b>C maj7</b>	C major7 CM7 CΔ	<b>C -7</b>	C min7 C minor7 Cm7 C mi7
<b>C 7/9/13</b>	C dom 7/9/13 C dominant 7/9/13 C 7/6/9 C dom7 6/9 C dominant 7 6/9	<b>C 7/13</b>	C dom 7/13 C dominant 7/13 C 7/6 C dom7/6 C dominant 7/6 C 13	<b>C -7b5</b>	C min7b5 C minor7b5 Cm7b5 CØ C half dim C half diminished	<b>C 7add4</b>	C dom7add4 C dominant 7add4 C 7/11 C 11 C dom7/11 C dominant7/11
<b>C +7</b>	C aug7 C augmented7 C 7#5	<b>C 7</b>	C dom7 C dominant 7 C dominant	<b>C 6</b>	C maj6 C major6 CM6	<b>C +</b>	C aug C augmented C #5
<b>C dim</b>	C diminished C°	<b>C 7b9</b>	C dom7b9 C dominant 7b9	<b>C 7b9b13</b>	C dom7b9b13 C dominant 7b9b13	<b>C 7b9/13</b>	C dom7b9/13 C dominant 7 b9/13
<b>C 7#9</b>	C dom7#9 C dominant 7#9	<b>C 7#9b13</b>	C dom7#9b13 C dominant 7#9b13	<b>C 7/9b13</b>	C dom7/9b13 C dominant 7/9b13		
<b>C 5</b>	(N/A)	<b>C dim7</b>	C diminished7 C°7	<b>C sus4</b>	C sus	<b>C sus2</b>	C 5add9 C 5add2
<b>C 6/9</b>	C maj6/9 C major6/9 CM6/9 CM6add9 C 6add9 C maj6add9 C major6add9	<b>C add9</b>	Cmaj add9 C major add9 CMadd2 CMadd9 C add2 C maj add2 C major add2	<b>C add4</b>	C maj add4 C major add4 CMadd4 CMadd11 C add11 C maj add11 C major add11	<b>C maj7/9</b>	C major7/9 CM7/9 CM7add9 CΔ9 C maj7add9 C major7add9 C maj7add2 C major7add2
<b>C -13</b>	C min13 C minor 13 Cm13 C -7/13 C min7/13 C minor7/13	<b>C -7b5/11</b>	C -7b5/11 C minor7b5/11 Cm7b5/11 C Ø11 C half dim 11 C half diminished 11	<b>C 7/9</b>	C dom7/9 C dominant 7/9 C 9 C7add9	<b>C 7#11</b>	C dom7#11 C dominant 7#11 C 7b5 C dom7b5 C dominant 7b5
<b>C -7/11</b>	C min7/11 C minor7/11 Cm7/11 Cm11 C -11 C min11 C minor11 C mi11 C mi7/11	<b>C min/maj7</b>	C -maj7 C minor/maj7 CmM7 CminΔ C-Δ C mi/maj7 C minor/major7 C minor major 7	<b>C aug maj7</b>	C +Maj7 C augmented major 7 C+M7 CaugM7 CaugΔ C+Δ C maj7#5 C major7#5 C+7 CΔ#5	<b>C -7/9</b>	C min7/9 C minor7/9 Cm7/9 Cm7add9 C min7add9 C minor7add9 C mi7/9 C mi7add9 C -7add9
<b>C maj7#11</b>	C major7#11 CM7#11 CΔ#11	<b>C 7sus4</b>	C dom7sus4 C dominant 7sus4 C 7sus	<b>C dim maj7</b>	C diminished major 7 CdimM7 Cdim7 C°M7	<b>C maj7/13</b>	C major7/13 CM7/13 CΔ13
<b>C -6</b>	C min6 C minor6 Cm6 C mi6	<b>C -b6</b>	C minb6 C minor b6 Cmb6				

credit: Ian Hersey

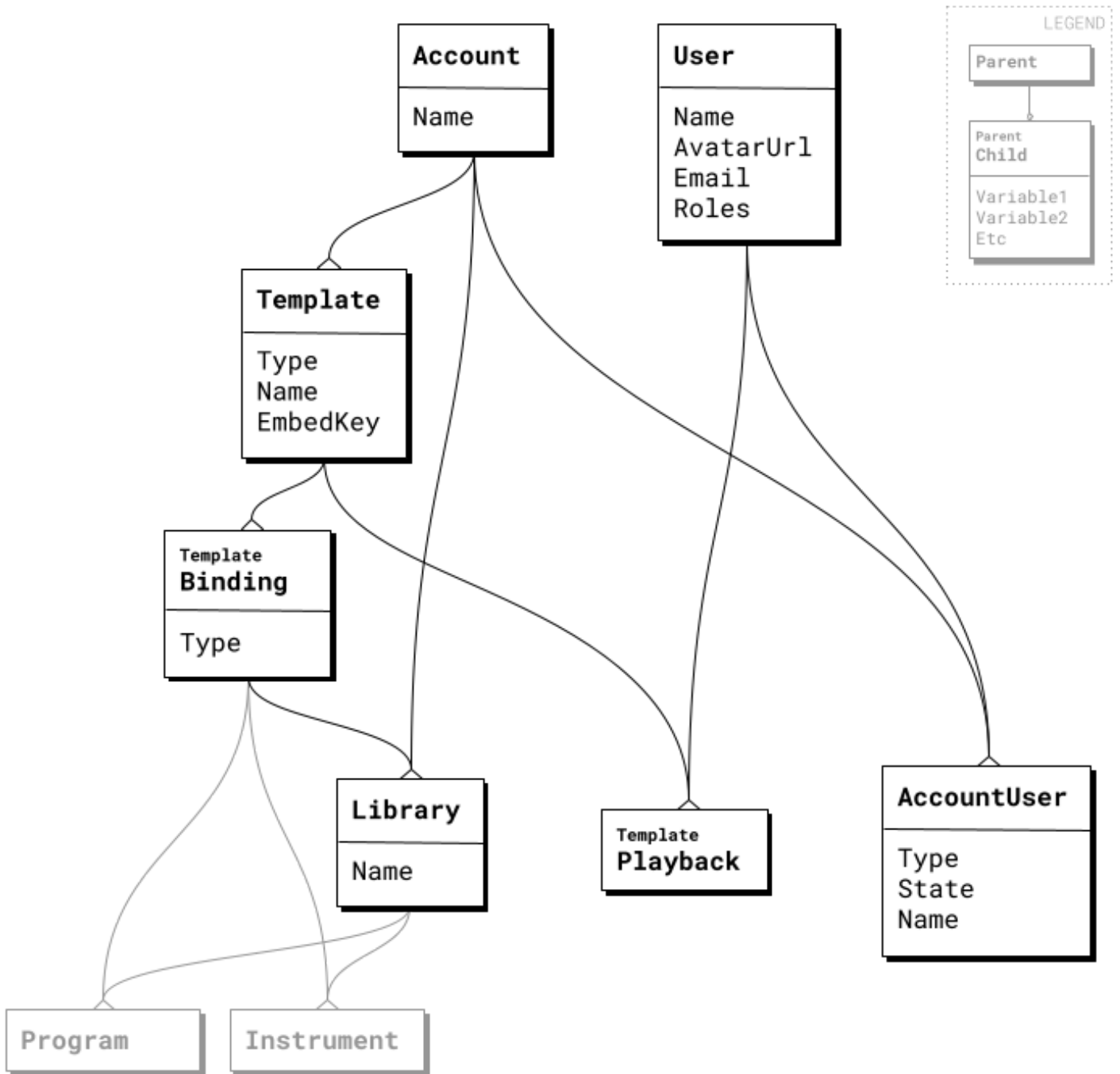
## Appendix B. Entities & their Relationships

We've discussed many entities & their relationships implicitly by showing all these screenshots of XJ music content libraries and describing the function of live music fabrication.

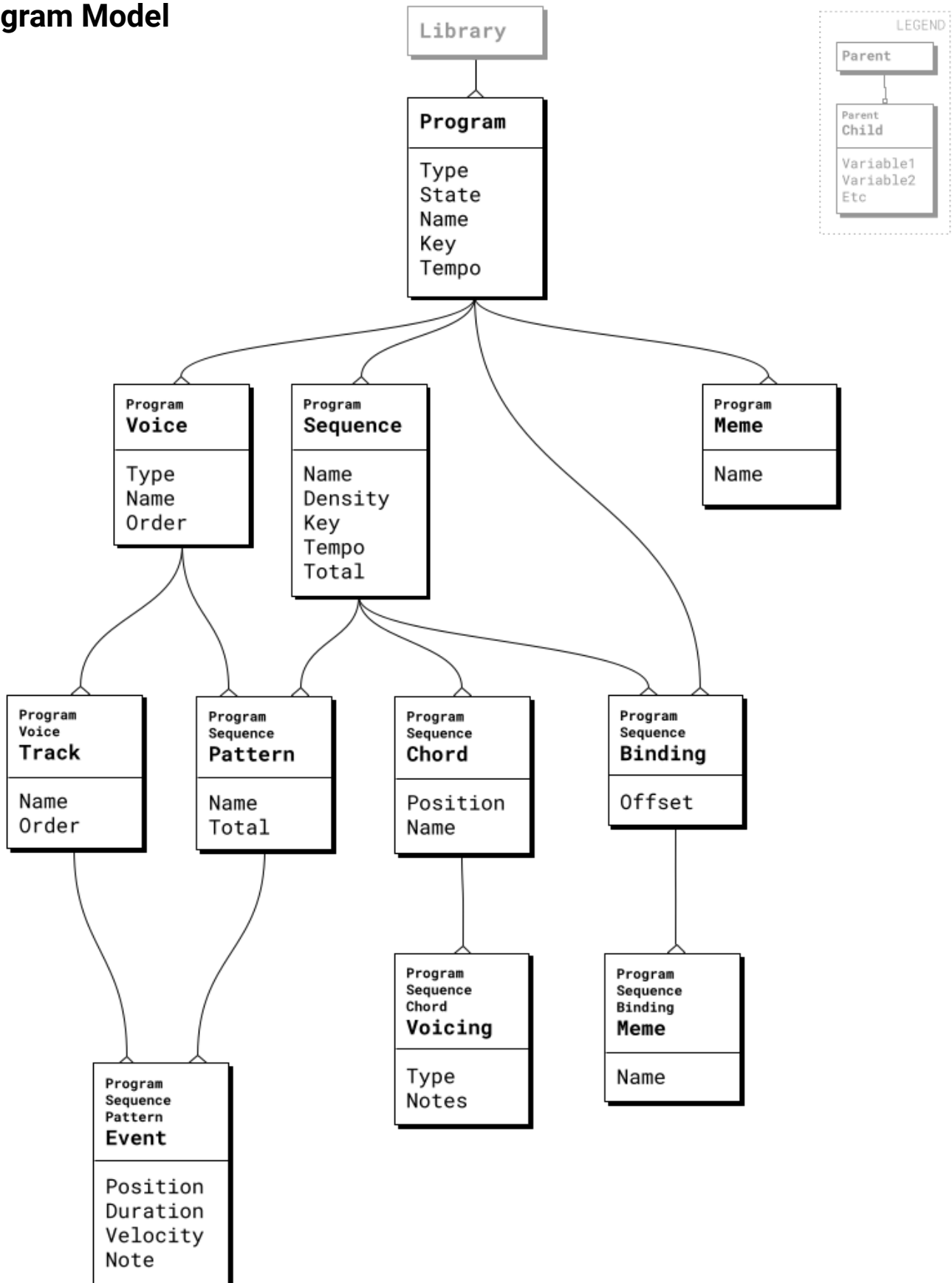
In this section, all entities are explicitly depicted by a series of Entity Relationship (ER) diagrams.



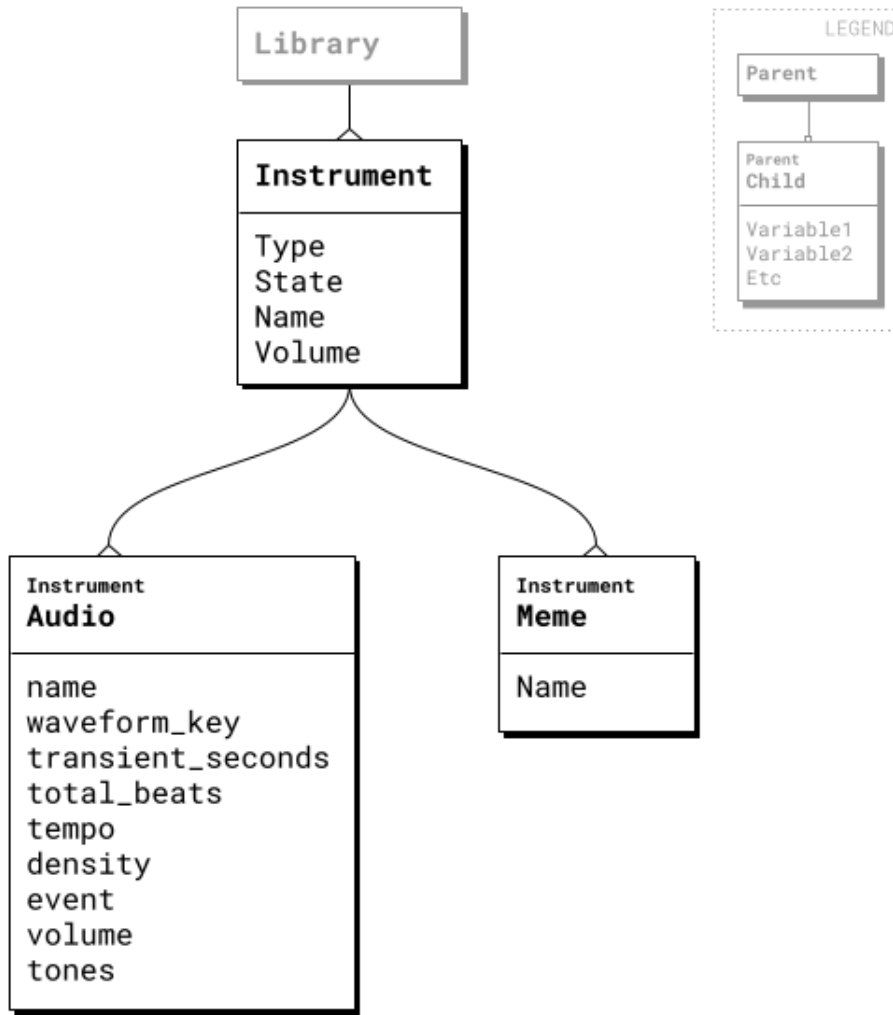
# Account Model



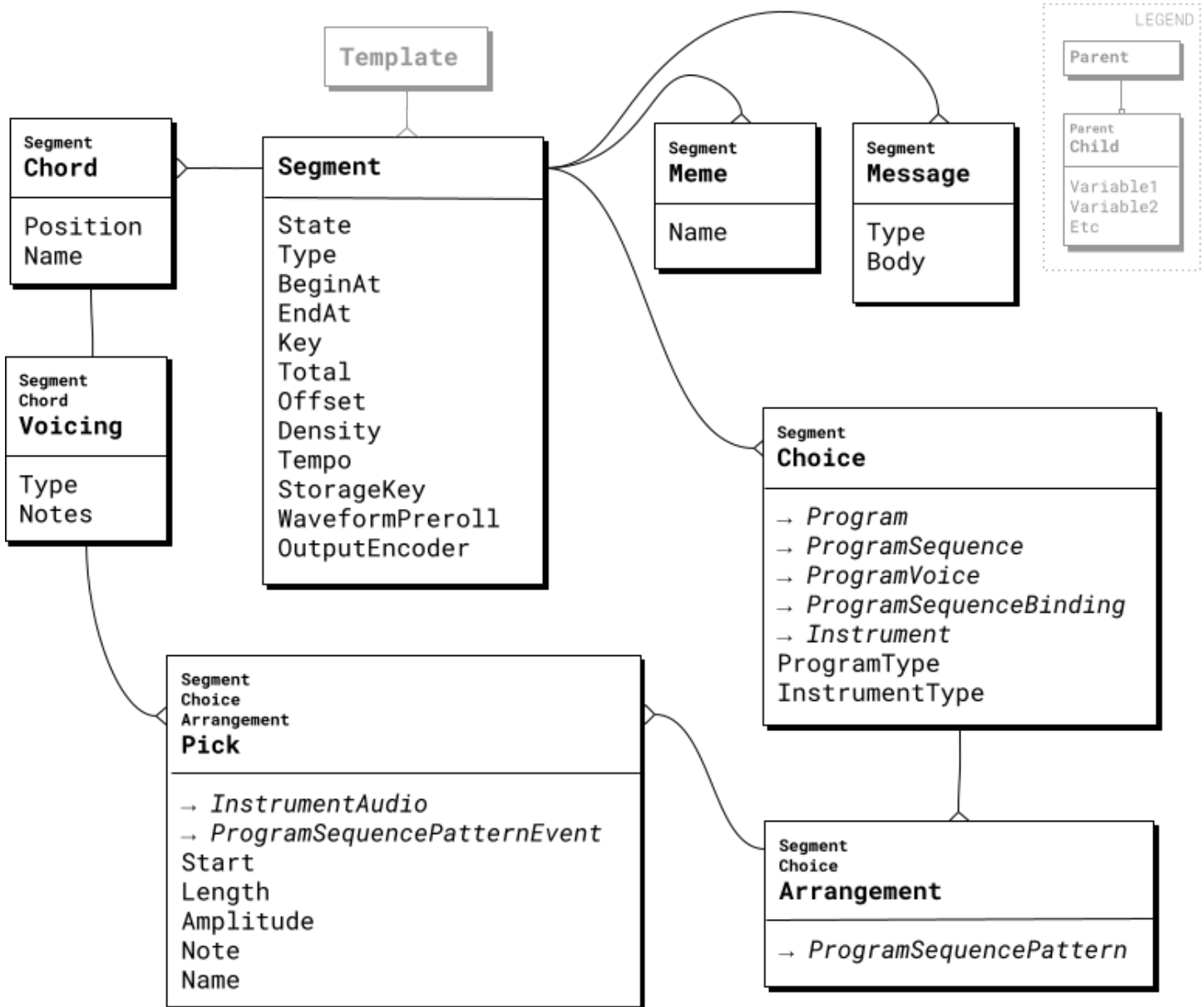
# Program Model



# Instrument Model



# Segment Model



# Glossary

## **Composition**

Artistic musical production showing study and care in arrangement. The act of composition is the process of forming a whole or integral, by placing together and uniting different parts.

## **Artist (Musician)**

A skilled practitioner in the art of composition and/or performance of music.

## **Engineer**

Person skilled in the principles and practice of music technology, including but not limited to audio engineering, and the operation of musical generation systems.

## **Digital Audio Workstation (DAW)**

Electronic device or software application used for recording, editing and producing audio files.

## **Audio Signal (audio data, audio sample, signal, audio, or sample)**

Information that represents audible sound, such as a digital recording of a musical performance, persisted in a file on a computer.

## **Generation**

A process by which data is created, including but not limited to recording the output of a microphone or performing complex mathematical operations.

## **Modulation**

A process by which data is modified in such a manner as to alter at least some property, including but not limited to the amplitude, frequency, phase, or intensity of an audible signal.

## **Configuration (config)**

The arrangement or set-up of the hardware and software that make up a computer system.

## **Audio channel (audio track, track, or channel)**

A single stream of audio data. Optionally, two or more channels may be played together in a synchronized group. For example, stereo output is comprised of a left channel and a right channel.

## **Audio composition (audio mixing or mixing)**

The process of forming new audio by placing together and uniting at least two source audio samples or channels. In the process, each source audio sample may be modulated such as to best fit within the composition of the final audio output.

## **Audio mixer (or mixer)**

An apparatus used to perform audio mixing.

## **Audio event**

An event which occurs at a specific position in time within a piece of recorded audio.

**Metadata**

Information describing musical properties, including but not limited to events, selections, notes, chords, or the arrangement of audio samples.

**Series**

Two items succeeding in order.

**Next**

Being nearest in time, or adjoining in a series. In an empty series, the next item would be the initial item added to the series.

**Terminus**

Either the initial or final item in a series.

**Static**

Having a permanently constant nature.

**Dynamic**

Having a changing or evolving nature.

**Permutation**

The arrangement of any determinate number of things, in all possible orders, one after the other.

**Note**

A musical sound, a tone, an utterance, or a tune. It may refer either to a single sound or its representation in notation.

**Pitch**

The frequency of vibrations, as in a musical note. The exact pitch of notes has varied over time, and now differs between continents and orchestras.

**Interval**

The distance in pitch between two notes. The violin, for example, is tuned in intervals of a fifth (G to D, D to A and A to E), the double bass in fourths (E to A, A to D and D to G).

**Harmonic intervals**

The distance between two notes which occur simultaneously, as when a violinist tunes the instrument, listening carefully to the sound of two adjacent strings played together.

**Melodic intervals**

The distance between two notes played in series, one after the other.

**Chord**

At least two notes played simultaneously at harmonic intervals.

**Scale**

At least two notes played in series at melodic intervals.

**Musical event**

An action having been, or intended to be performed by a musical instrument, beginning at a specific moment in time, continuing for some amount of time, having characteristics including but not limited to chord, pitch, or velocity.

**Harmonic event**

A single occurrence of an action having been, or intended to be performed by a harmonic instrument.

**Melodic event**

A single occurrence of an action having been, or intended to be performed by a melodic instrument.

**Harmonic progression**

The placement of chords with relation to each other such as to be musically correct and emotionally evocative.

**Key (root key or key signature)**

A musical composition indicating the scale to be used, and the key-note or home-note. Generally, a musical composition ends—evoking resolve—on the chord matching its key. The key of a musical composition determines a context within which its harmonic progression will be effective.

**Voice**

A single identity within a musical composition, such as might be performed by a single musical instrument. A voice is either percussive, harmonic, or melodic.

**Voice event**

A single occurrence of an action having been, or intended to be performed by a single voice of a musical composition. An event has musical characteristics, representing a particular note or chord.

**Song**

A musical composition having a beginning, a middle, and an end.

**Section**

A distinct portion of a musical composition.

**Partial musical composition (part)**

A subset of a complete musical composition, such as to be interchangeable with other subsets of other compositions.

**Composite music**

A work of musical art created dynamically from distinct parts or elements, distinguished from traditional recorded music, which is mastered and finished statically as a deliverable record.

**Aleatoric music (music composed “aleatorically”)**

Music in which some element of the composition is left to chance, and/or some primary element of a composed work's realization is left to the determination of its performer(s).

**Program (musical program or main program)**

A partial musical composition comprising or consisting of a progression of chords and corresponding musical

events output of said related thereto and/or represented by stored musical notations for the playback of instruments. A program is comprised of at least some section representing a progression of musical variation within the program.

**Composite musical program**

An integral whole musical composition comprised of distinct partial musical programs.

**Macro Program**

This is like a playlist. Musical notation describing the order in which to play Main Programs.

**Main Program**

This is like a song. Musical notation describing chord progressions and voicings of those chords for various types of instruments.

**Beat Program**

This describes a beat. XJ chooses one beat for an entire song (Main Program). This musical notation is used directly for a Drum Instrument to play. The beat is also used to make choices about other arrangements.

**Detail Program**

This is writing down the rhythm of a part for one instrument to play, in a way that may be interpreted by XJ in for any song according to its chord voicings. The most atomic and portable sort of partial musical composition, and is intended to be utilized wherever its musical characteristics are deemed fit.

**Instrument**

A collection comprising or consisting of audio samples and corresponding musical notation related thereto and/or represented by stored audio data for playback.

**Library**

A collection consisting or comprising of both programs and instruments, embodying a complete artistic work, being musical composition which is intended by the artist to be performed autonomously and indefinitely without repetition, by way of the present disclosed technology.

**Chain**

An information schema representing a musical composite.

**Segment**

An information schema representing a partial section of a chain. A chain comprises a series of at least one segment.

**Meme**

The most atomic possible unit of meaning. Artists assign groups of memes to instruments, programs, and the sections therein. During fabrication, entities having shared memes will be considered complementary.

**Choice**

A decision to employ a particular program or instrument in a segment.



**Arrangement**

The exact way that an instrument will fulfill the musical characteristics specified by a program. This includes the choice of particular audio samples, and modulation of those audio samples to match target musical characteristics.

**Node**

A term commonly used in the mathematical field of graph theory, defined as a single point.

**Edge**

A term commonly used in the mathematical field of graph theory, defined as a connection between two nodes.

**Morph**

A particular arrangement, expressed as nodes and edges, of audio samples to fulfill the voice events specified by a program.

**Sub-morph**

A possible subset of the events in a morph.

**Isometric**

A term commonly used in the mathematical field of graph theory, defined by pertaining to, or characterized by, equality of measure. Set A and Set B are isometric when graph theoretical analysis finds similar measurements betwixt the items therein.

**Audio event sub-morph isometry**

The measurement of equality between all sub-morphs possible given a source and target set of audio events.

**Time-fixed pitch-shift**

A well-known technique used either to alter the pitch of a portion of recorded audio data without disturbing its timing, or conversely to alter its timing without disturbing the pitch.

**Artificial Intelligence (AI)**

The theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and music generation.

## References

Kaye, N. Charney. (2019). *System for generation of musical audio composition* (U.S. Patent No. 10,446,126). U.S. Patent and Trademark Office. <https://patents.google.com/patent/US10446126B1>

# Authors

Charney Kaye

Mark Stewart

Jamal Whitaker

Ian Hersey