

Understanding Antigone's architecture

Introduction

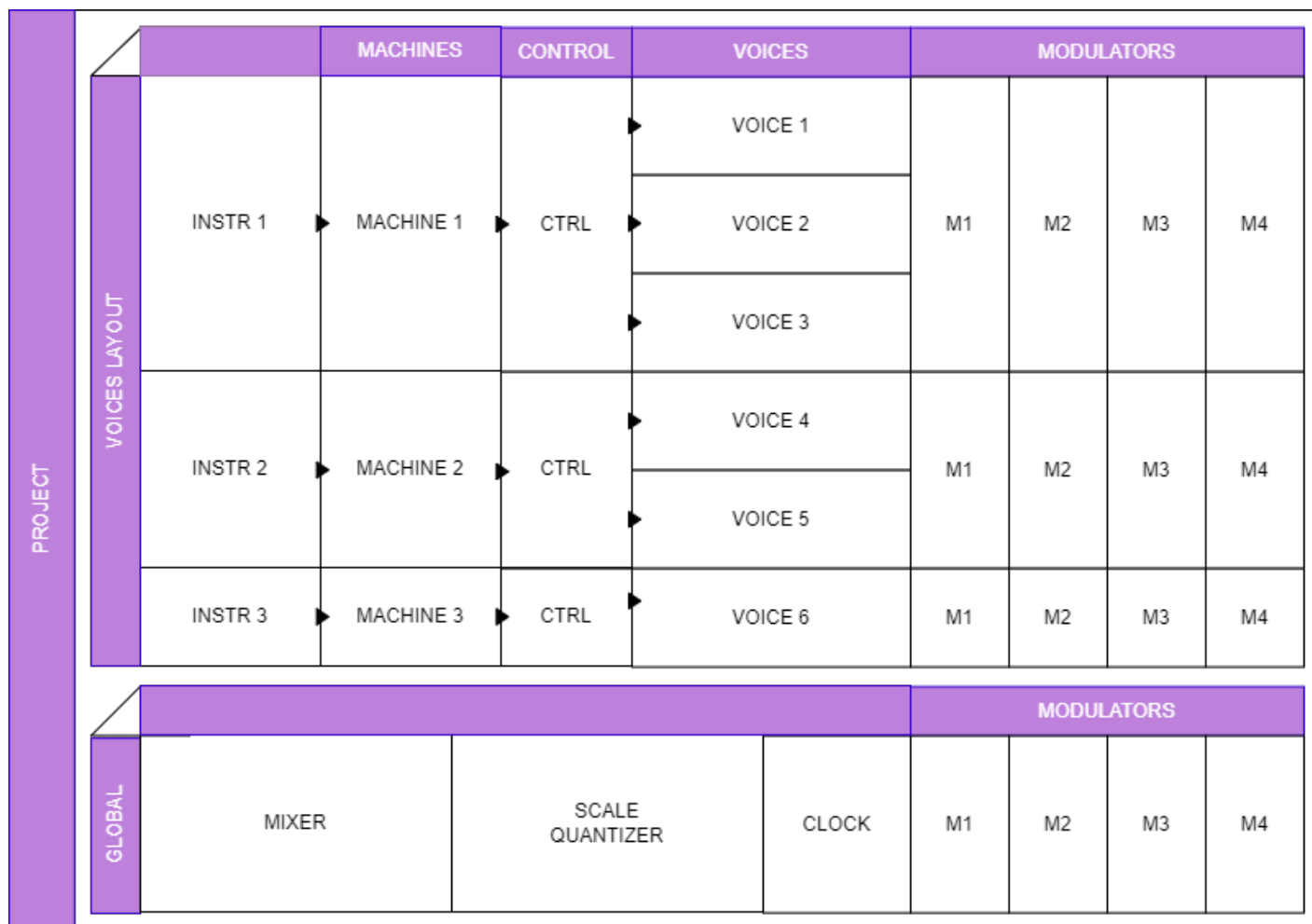
Antigone offers an architecture that may seem complex, but it is ultimately quite simple to understand if you take the time to explore it.

The Projects

The concept of a project is very similar to the projects offered by Elektron machines (such as the Digitakt, ...). The project is the foundation of everything within Antigone.

Projects can be compared to presets, patches, or programs found in some other modules or synthesizers.

Here is the general schematic of a sample project that we will refer to multiple times in this section of the documentation. Keep an eye on it as you read.



A project consists of 2 main sets organizing 3 major concepts:

- **The VOICES LAYOUT** which organizes **the INSTRUMENTS**
- **The GLOBAL** (or global parameters)

While the **VOICES LAYOUT** defines how the **INSTRUMENTS** of your project are distributed across all available voices, **GLOBAL** defines the common behavior of all **INSTRUMENTS** in a given project.

The Concept of Voice: The Voices Layout

Before diving into how the **INSTRUMENTS** and **GLOBALs** work, let's take a moment to understand the concept of voices.

Antigone offers a maximum of **6 voices per project**. Voices are the circuits capable of emitting sound. Being able to use 6 voices means, for example, that it will be possible to create a 6-voice polyphonic instrument (capable of playing 6-note chords) or to create 6 monophonic instruments (each instrument can only play one note at a time), and all possible intermediate combinations (e.g., 2 instruments with 3 voices each, 3 instruments with 2 voices each, etc...)

In the above diagram, you can see an example of a project consisting of 3 instruments, each distributed with 3, 2, and 1 voice respectively.

The Instruments

A project can contain from 1 to 6 instruments depending on how they are distributed across the 6 available voices (as defined by the VOICES LAYOUT)

An instrument is composed of 3 different components

Components	Role	Learn More
a machine	<p>an algorithm capable of generating sound using a specific sound synthesis.</p> <p>At the time of writing this documentation, there are 4 different types of machines, each offering very different sounds, giving Antigone a wide range of sound palettes.</p>	<i>coming soon</i>
a controller	whose parameters define how the instrument behaves when you control it	<i>coming soon</i>
4 modulators	<p>capable of running in parallel, these modulators will allow you to complicate the behavior of their instrument over time.</p> <div><p>Although common to an instrument, you will be able to define specific modulation parameters for each of its voices. This makes Antigone very, very powerful.</p></div>	<i>coming soon</i>

Global: The Global Parameters

The global parameters define the overall behavior of your project.

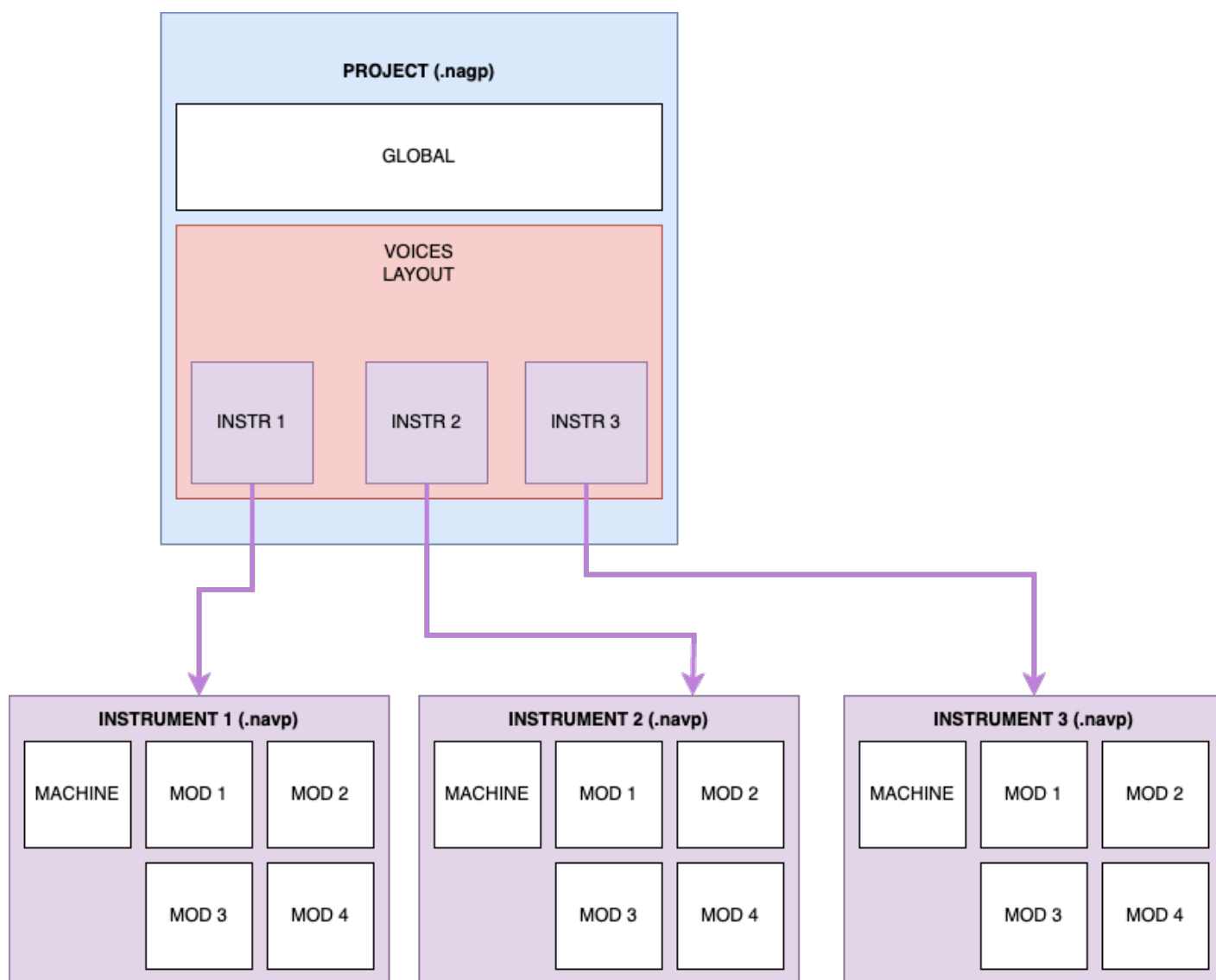
A global is composed of 5 different components

Components	Role	Learn More
------------	------	------------

a mixer	A mixing table defining the behavior of each instrument in the project at the audio level: volume, pan	<i>coming soon</i>
a scale quantizer	This component confines the notes played within a defined scale and transposed pitch.	<i>coming soon</i>
a clock	This module allows you to define the clock parameters used to synchronize the module's behaviors. This clock can be either internal or external to the module.	<i>coming soon</i>
4 global modulators	Capable of running in parallel, these modulators will allow you to complicate the overall behavior of the project over time.	<i>coming soon</i>
A modulation matrix	This component allows you to visualize the entirety of a project's modulations within a matrix.	<i>coming soon</i>

File Content by Type

Now that we know how the different components of a project in Antigone are organized, it is important to understand how these same concepts are structured as files on your SD card.



For the 3 concepts mentioned above (PROJECT, VOICES LAYOUT, and INSTRUMENT) there is a file type.

Concepts	File Content	File Type
Project	<p>All the components that define it (the global parameters, the voice layout used, and the configuration of each instrument).</p> <p>It is, therefore, complete and autonomous.</p>	.nagp
Instrument	<p>A machine and its modulations.</p> <p>You can use this file type to save a reusable instrument in another project.</p>	.navp

Revision #6

Created 1 September 2024 13:44:55 by Pascal

Updated 19 April 2025 07:20:36 by Vincent