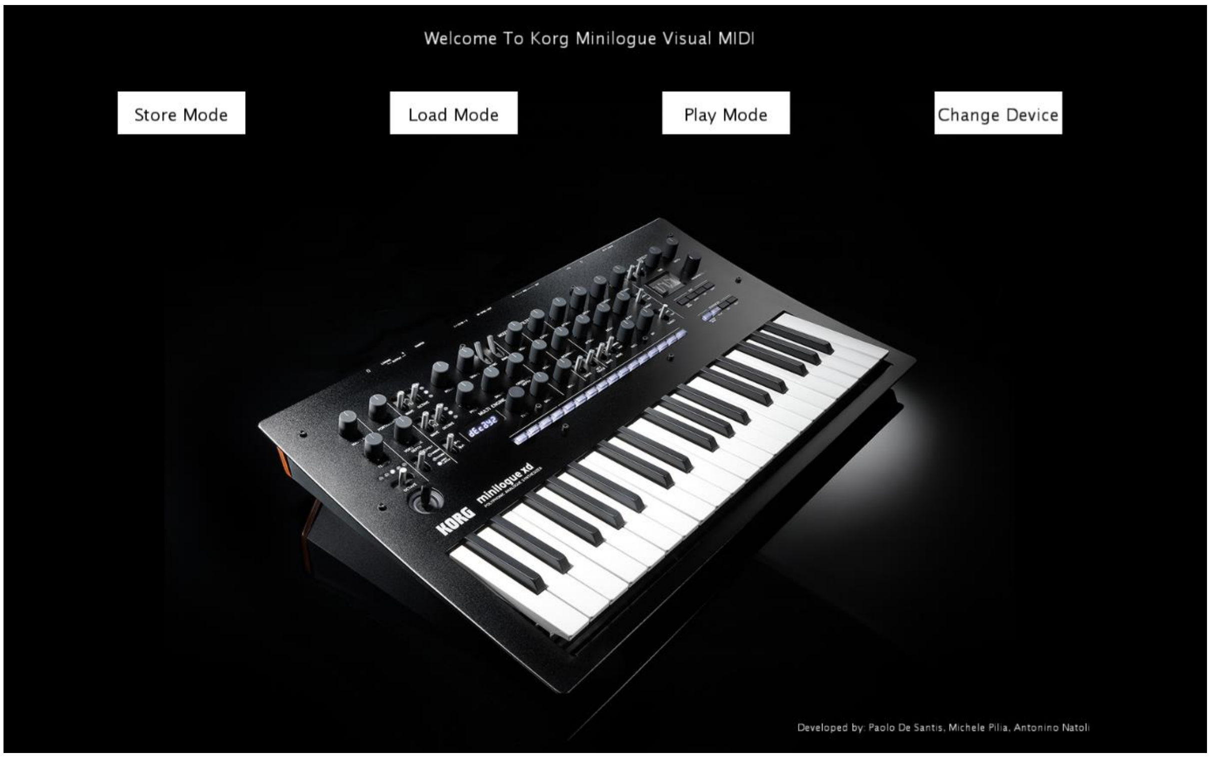
# Visual Midi

Developed by Paolo De Santis, Michele Pilia, Antonino Natoli, for the “Creative Programming And Computing” course, M.Sci. in “Music And Acoustic Engineering”, at Politecnico di Milano.


Welcome To Korg Minilogue Visual MIDI
Store Mode
Load Mode
Play Mode
Change Device
KORG
minilogue xd
Developed by: Paolo De Santis, Michele Pilia, Antonino Natoli

GitHub Repository:
https://github.com/DesaPaolo/VisualMidi---Creative-Computing-Project

Java Doc:
https://drive.google.com/file/d/1Cb4k5xCztftsOunxKK11CRqIr2cnQHdH/view?usp=sharing

Demo Video:

# 1 Introduction

## 1.1 Our Idea

Our idea is to design a creative tool, plug and play, for a visual representation of MIDI messages transferred from a Korg Minilogue, one of the most common analog synthesizer. This visual representation should be coherent with the sound features perceived and modulated in real time.

## 1.2 Our Goal

Our goal is to allow to a performer or a band, during a live event, to have the possibility of having a great visual representation of what is happening on the sonic plane. Imagine a big LCD screen that displays a visual concept that is coherent with the sound source.

## 1.3 Why Korg Minilogue?

We have chosen the Korg Minilogue as our standard MIDI device because it's one of the most common analog synthesizer, great sounding and easy to use.

# 2 Overview

## 2.1 Core Features

In this section we list all the main features of the application.

- Main menu
  - This is the screen displayed once you have launched the program. It features a Minilogue image and several buttons on top of it. By clicking one of them the program navigates to the corresponding modality.
- Store Mode
  - Allows a user to store a Minilogue preset, to be retrieved during a live performance.
- Load Mode
  - Allows a user to load a Minilogue preset. It allows a user to see the programs names and values of the stored presets.
- Change Device Mode
  - Allows a user to change the device driver of the Minilogue and of the Kemper Profiler. In this way you can also map your own controller, or use a pre recorded midi clip. In case you use another midi controller you'll loose all the coherence between sound and video.
- Play Mode
  - Here is where the magic happens. Just play notes, tune knobs. The graphics updates according to the received MIDI messages. During play mode you can retrieve already saved Minilogue presets just by rotating the "PROGRAM/VALUE" knob to the desired preset.

- The main graphical elements and effects featured in this mode are: spheres, spirals, particle systems and a star field.

## 2.2 Programming Language Used

We used Processing with 3D rendering, since it was the programming language required by the exam and also because of his great power in bringing great graphics.

# 3 Graphical Parameters

In this chapter we list all the sound features we implemented both for the Minilogue and the Kemper Profiler. In the development we used a lot the Particle System technique to achieve a great representation.

## 3.1 Minilogue Parameters

We list the timbre features associated to their graphical representation

- Notes played on the keyboard: spheres with a position on the screen that depends on the pitch
- Pitch Bend: stretching the size of the spheres
- LFO Pitch: Moving the spheres up and down
- Filter Cutoff: background color and spheres size
- ADSR envelope: opacity and z-axis position of the spheres
- Filter ADSR envelope: modulating the background color and spheres size
- Delay: displays a particle system
- Delay Feedback: life of the particle system
- Delay Time: rate of born of particles

## 3.2 Kemper Profiler Parameters

As regarding the Kemper Profiler, SysEx messages are sent to the software. In the software there is an algorithm that converts the parameters received into intelligible attributes. These attributes with their possible values are listed here:

- Amplifier Type: Clean (default), Crunch, Hi Gain.
- Overdrive Type: None (default), Overdrive, Boost, Distortion, Fuzz.
- Modulation Type: None (default), Phaser, Flanger, Chorus.
- Equalizer Type: Normal (default), Warm, Bright.
- Reverb Type: Small (default), Medium, Large.

Now we list the graphical effects associated to the above parameters

- Amplifier Type: affects the stars track colors and size
- Overdrive Type: affects the stars colors and their perceived motion direction
- Modulation Type: if is different from "none" enables a rotation of the sphere and the display of  spirals, each orbiting  around a sphere.
- Equalizer Type: affects the color of the spheres and of the particle systems associated to the delay. The color can be extracted from a pool of standard colors with an uniform probability or by using a Perlin noise.
- Reverb Type: affects the density of the stars in the starfield

# 4 Software Documentation

This section contains a brief documentation of software design choices we made.

## 4.1 External libraries and global variables

External libraries and global variables are imported and declared into a specific file.

As regards external libraries we used (excluding Java ones)

- Midi Bus: was used in order to retrieve MIDI messages from the Minilogue.
- CoreMidi4J: was used to retrieve SysEx messages from the Kemper Profiler, due to iOS compatibility problems.

## 4.2 Setup function

This function is called only one time, once the program is launched. It sets the size of the sketch, and it loads the initial page, with the Minilogue image and the four buttons on top of it.

## 4.3 Draw function

This function is called each frame and updates the graphics based on the model variables. Many of the classes has their own draw method, which is called by this draw function each frame.

## 4.4 Java Doc

We have also generated the Java Doc documentation using Doxygen. The latter contains all the details of the classes, methods that we used. You can download it by clicking here.