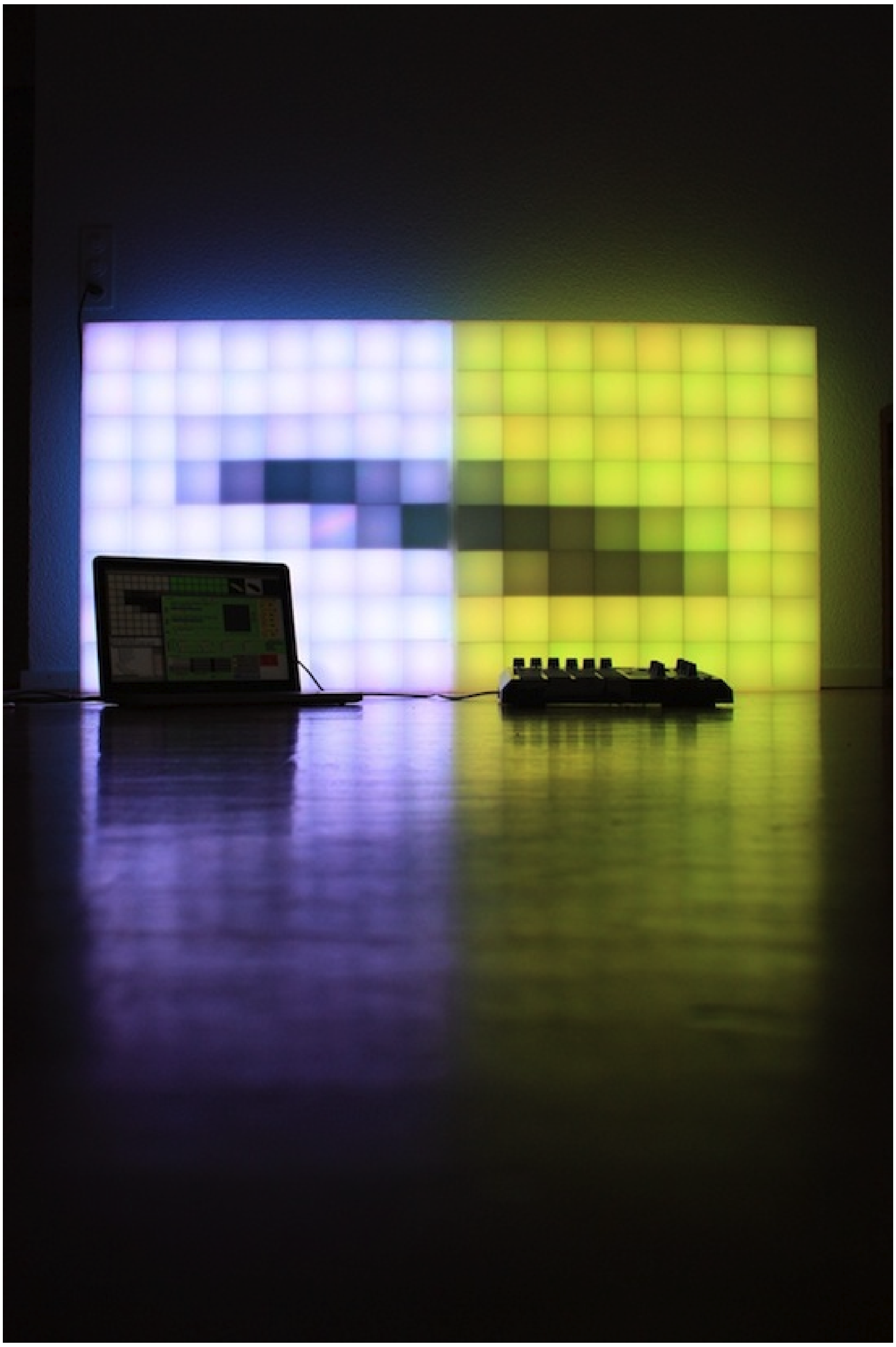


PixelInvaders Manual

Intro

PixelController is the control software for the PixelInvaders panels. The software is written in Java and therefore platform independent. PixelController was tested under Windows, Linux and Mac OSX.

The software is released under the GPLv3 open source license and isat availablehttps://github.com/neophob/PixelController.

Required software

PixelController needs some 3rd party applications:

- Java Runtime v1.6 (JRE http://java.com/de/)
- Pure Data Extended 0.42.5(http://puredata.info/community/projects/software/pd-extended)
- Serial port driver (http://www.pjrc.com/teensy/td_download.html, “Windows Serial Installer”),
- possibly WinZip or similar software.

Pure Data Extended serves as a GUI front-end. say to the operation of the application. WinZip (or similar) is required for installation.

Installing and starting PixelController

If the required software components are installed, PixelController can to be unpacked. For this purpose, the supplied zip file unpacks into a directory, eg for "c:\PixelController\".

To start the application is the batch file "c:\PixelController\PixelController.cmd" started.

PixelController Concept

PixelController generates real-time animations, which are displayed on one or more LED panels. An generated animation is called ***"Visual"*** and looks like this:

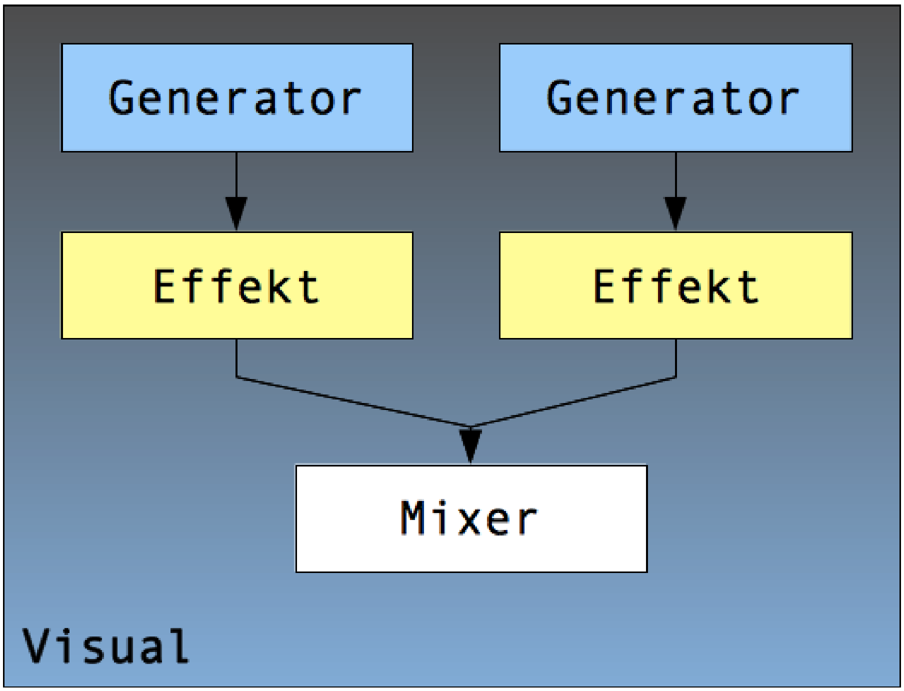


Illustration 1: definition of a visual

A ***generator*** produces an image content, for example a image or video is displayed. The ***effect*** will alter the visual content, for example, the image content becomes red or rotated. Two such image content eventually merged with a ***mixer,*** for example the contents are added or multiplied.
A Visual defines what is displayed but not How and Where. That where the ***mapping*** comes into the game.
A visual can eventually be displayed on one or more panels such as the following example shows:

Figure 2: Each Panel displays a visual.

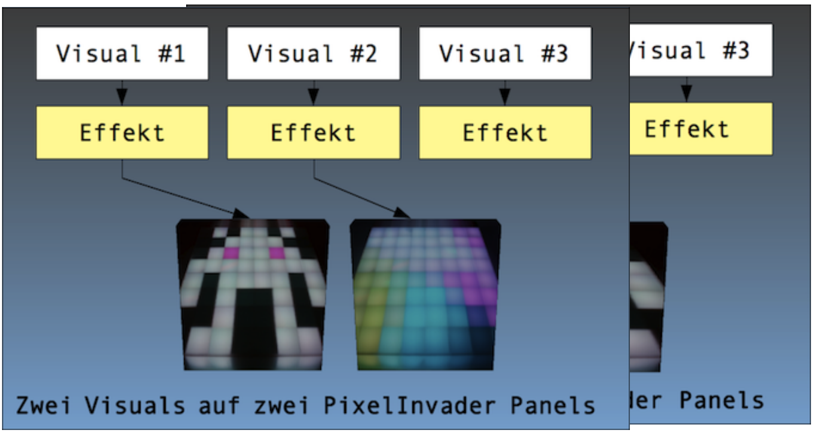
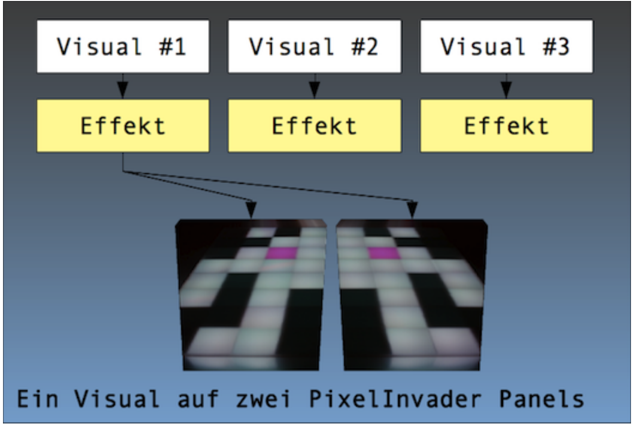


Illustration 3: A Visual is stretched on two panels.



PixelController configuration

In the PixelController datadirectory (eg "c:\PixelController\data"), the basic configuration of the application and media data such as images, fonts ... can be found.

Here is a sample configuration of PixelController (excerpt):

```
#=========================
#default values for fx
#=========================
initial.image.simple=logo.gif
initial.image.zoomer=bnz20.jpg
initial.blinken=torus.bml
initial.texture=logo.gif
initial.text=PIXELINVADERS

font.filename=04B_03__.TTF
font.size=82

#x/y offset for screen capturing generator
screen.capture.offset=100
```

Here are the initial values of various generators defined, e.g. as a standard image "logo.gif" is defined.

```
#=========================
#settings for PixelInvaders Panel, valid options:
#       NO_ROTATE,
#       ROTATE_90,
#       ROTATE_90_FLIPPEDY,
#       ROTATE_180,
#       ROTATE_180_FLIPPEDY,
#       ROTATE_270,
#=========================
#pixelinvaders.layout.row1=ROTATE_180,NO_ROTATE
#pixelinvaders.layout.row2=NO_ROTATE,NO_ROTATE

#=========================
# optional, defines the color order of the device
# this option is used for ALL output devices
#=========================
panel.color.order=RGB,RBG
```

The PixelInvaders panels gets configured here. Depending on how the panels are assembled, were the image must be rotated or flipped. If a second layout defined line (layout.row2), this means that there are several panels on the vertical axis.
The Color Order defines whether a panel uses RGB or RBG pixels, by default RGB is assumed.

```
#=========================
#network port config
#=========================
net.listening.port=3448
net.listening.addr=127.0.0.1
net.send.port=3449

show.debug.window=true
led.pixel.size=30
```

"Net.*" defines the network ports that are needed for communication with PureData.
"Show.debug.window" determines whether a simulated panel output is shown on the screen.
"Led.pixel.size" defines how big the should be pixels on the screen.

PixelController operation

Once the controller GUI was started (double clicking the file or PixelController.pd or start PureData, then “File open”) should be presented the following picture:

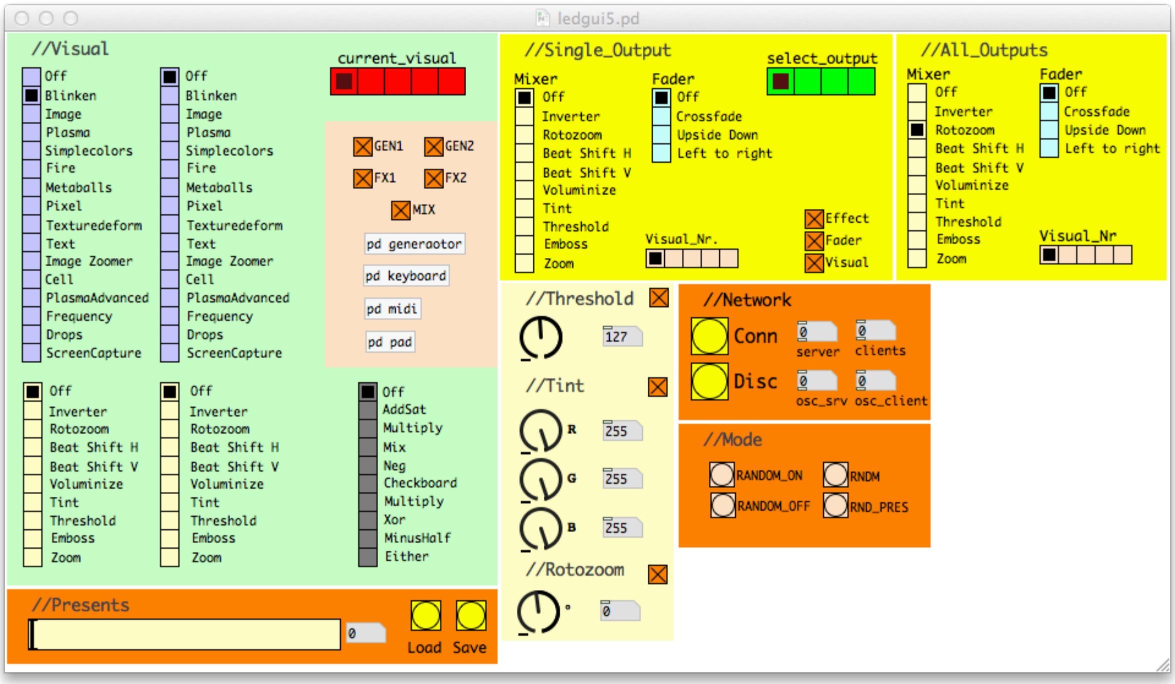


Illustration 4: Pure Data Interface

Network connection

The GUI must be connected to the PixelController application. This is what the "Network" section. If the automatic connection does not work, can be connected with the application by clicking on the "Connect" button, the GUI.

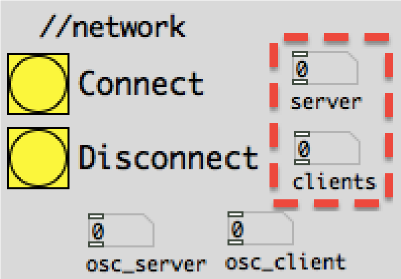


Illustration 5: No connection with the application

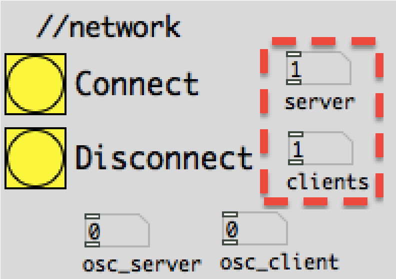
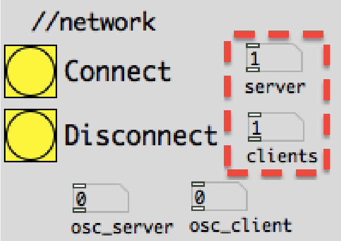


Figure 6: Combined with the application



Modes

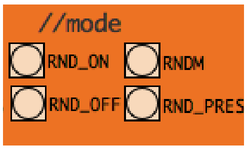


Illustration 7: Operating Modes

PixelController supports different modes of operation. When the application starts, PixelController is in manual mode, that is, each change must be done manually.

Clicking on ***"RND_ON"***activates the random mode. The screen will now be changed continuously, without having to intervene. However, it can be defined ***which*** functions are changed. See "Illustration 5", the effect is never changed in this case, however, the faders and the Visual.
Clickingon ***"RND_OFF"***disables the random mode.

A Click on ***"RNDM"*** is a unique action, all settings are chosen randomly.

A Click on ***"RND_PRES"*** is also a one-time action, here is an already saved configuration loaded.

Control Visuals

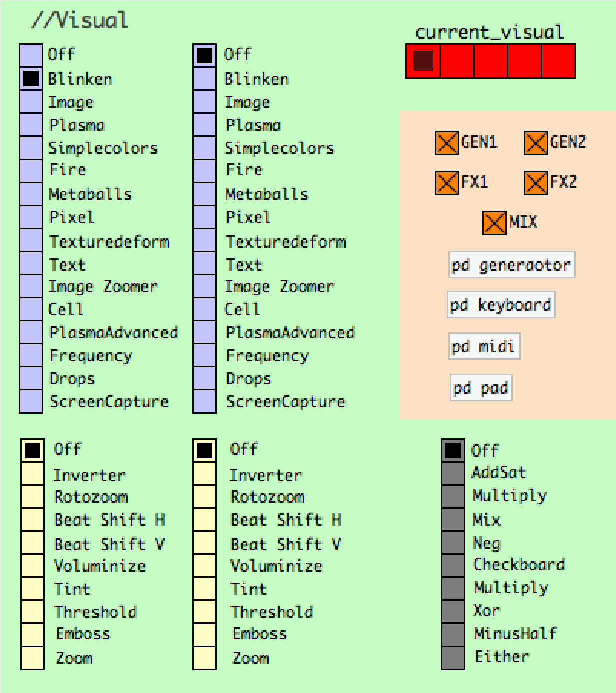


Illustration 8: Visuals control

The "current_visual" defines the active Visual. There is always one more visual than physical panels. This allows one to play with the settings and experiment.

A Visual consists of two generators, each with one effect which are mixed together. Here, these parameters can be defined. Has to be said is that the first option is always defined as an off / Pass-through is.

Displays Visuals on panels

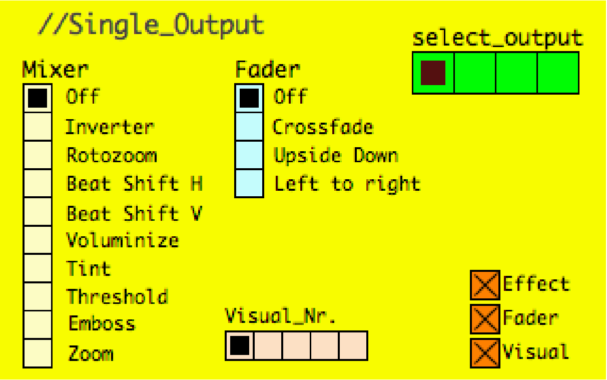


Illustration 9: Visuals folders andthread

An ***effect*** can be assigned to each panel (e.g., the image is inverted). In addition, each output gets a ***fader***.

A ***fader*** allowing smoother transitions when, for example, the visual is changed. Each output must be assigned to a Visual. If a ***visual*** display on multiple panels, the image content will be adjusted accordingly and the image content appears on several panels.

Visual Settings Save and Load

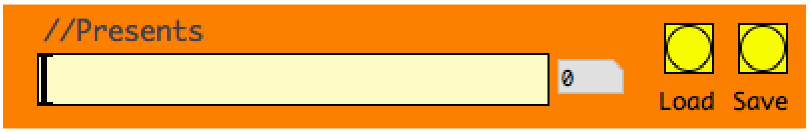


Illustration 10:Presents, save and load settings

It is possible to save the current configuration of the visuals and later to load it again. You simply select the memory from slot (will then be shown when the "current_slot" display) and save / load the appropriate configuration by clicking the appropriate button.

Effect settings

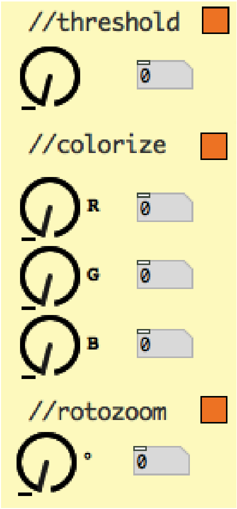


Illustration 11: Effect settings

Some effects, such as "Threshold", "Colorize" and "Rotozoom" have parameters that affect the functioning. These parameters canwith the slider from Illustration1 be changed1.

Generator settings

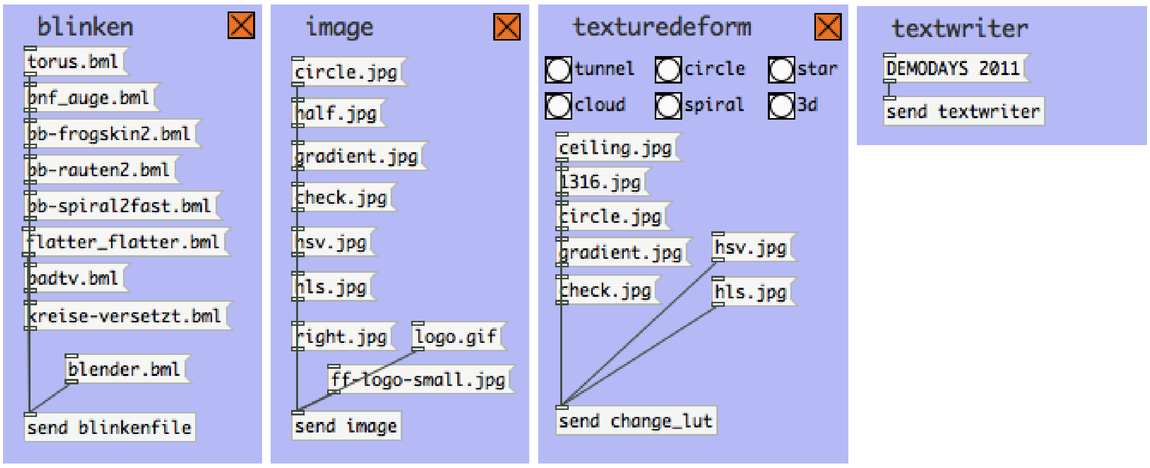


Illustration 12: Generator settings

After clicking on "pd conditions" (Misc section in the GUI) shown above, the mask are displayed. Here, the parameters of the generators are subject to change. The parameters displayed options (eg circle.jpg or badtv.bml) represent files that must exist in the data directory!

Additional Information

The above shows how far from the whole potential, which in Pure Data plugged. It is possible to control the application with a MIDI device (see "pd midi" subpatch) or with an OSC[1] application.

References:

- PixelInvaders Website: http://www.pixelinvaders.ch
- PureData Website: http://puredata.info/
- Source code of

[1] OSC Open Sound Protocol