Katapult Documentation

Introduction

Thank you for downloading Katapult! This document will describe setting up the application and using it with your software or hardware.

Contents

Controlling Katapult 2

Getting started 3

Creating Layouts 7

- The basics 7
- Control types 8
- Other entries 10
- Colors 12
- Keys 12

MIDI Devices 13

- For Mac 13
- For Windows 14

Bi-directional communication 15

MIDI Output Format 15

MIDI Input Format 15

Using Katapult alongside Ableton Live 15

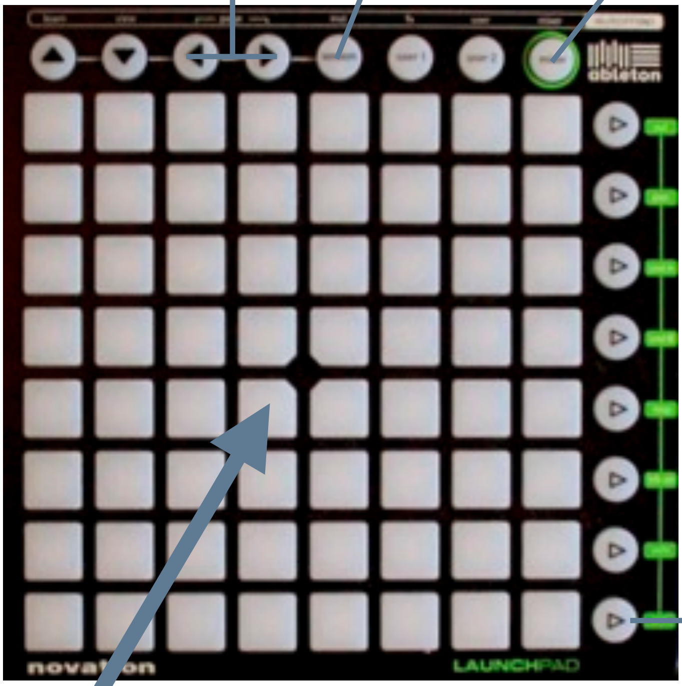
Mapping the round side-buttons 15

Troubleshooting 16

Contact information 16

Controlling Katapult

On Launchpad



Flip pages

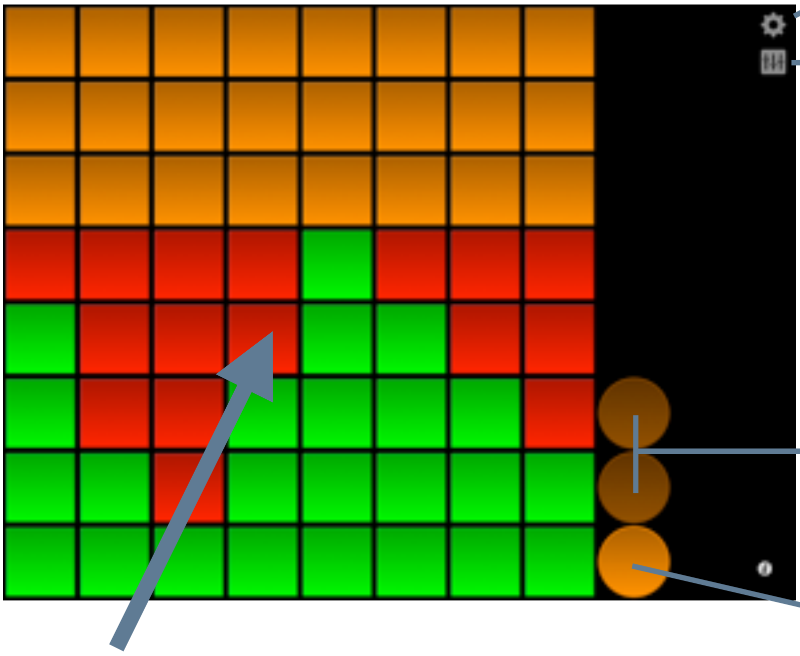
Reload layout

“Screen-saver” mode

Page selector

Your layouts here :)

On iPad



Settings

Connection indicator

Flip pages

Page selector

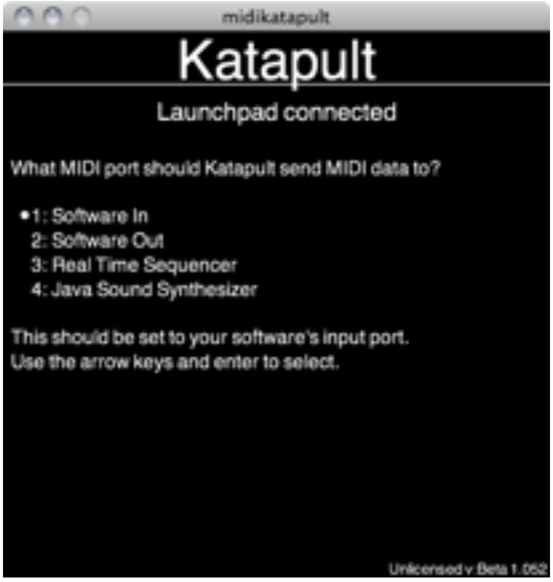
Your layouts here :)

Getting started

This section will guide you through starting up Katapult for the first time. If you are having trouble at any point through this guide, please refer to the “Troubleshooting” section. If that doesn’t solve your problems, feel free to email me, and I’ll help you out.

Let’s get started!

First of all, make sure that your Launchpad is connected by USB to your computer, then start the application. If you’re using an iPad, just go ahead and launch the application. When Katapult is started, you will presented with the following screen:



Katapult

Launchpad connected

What MIDI port should Katapult send MIDI data to?

- •1: Software In
- 2: Software Out
- 3: Real Time Sequencer
- 4: Java Sound Synthesizer

This should be set to your software's input port.
Use the arrow keys and enter to select.

Unlicensed v Beta 1.052

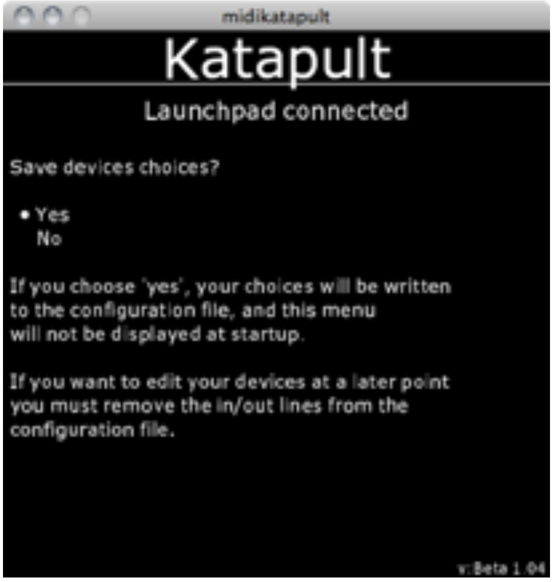
This screen lists all the available MIDI devices on your system. Select what MIDI device you want to send MIDI to from the menu and press enter. You should select the MIDI device that your software uses. If your software doesn’t publish it’s own MIDI device, you can create a virtual MIDI device yourself. See the “MIDI Devices” section for a guide on how to do this.

Tip! *The Katapult output device is the device that Katapult* *<u>transmits</u>* *data to. This should be set to the device your software uses to listen for incoming data.*

The MIDI data flows like this:

Katapult -> Output Device (in this example named “Software In” -> Software

When you have selected your output device in Katapult, you will be presented with the following screen:



midikatapult

Katapult

Launchpad connected

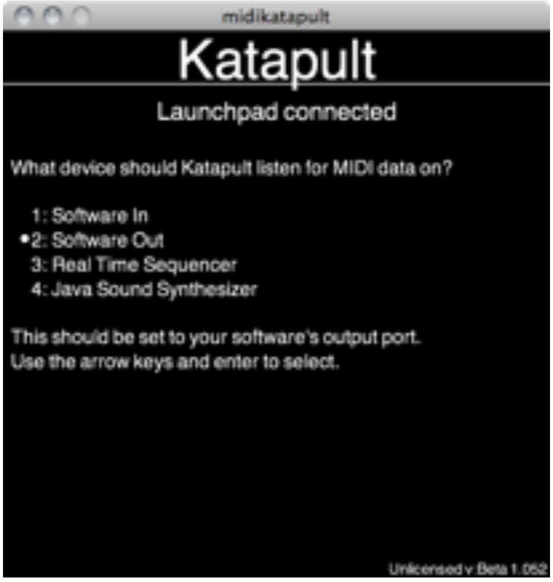
Save devices choices?

- • Yes
- No

If you choose 'yes', your choices will be written
to the configuration file, and this menu
will not be displayed at startup.

If you want to edit your devices at a later point
you must remove the in/out lines from the
configuration file.

v:Beta 1.04



midikatapult

Katapult

Launchpad connected

What device should Katapult listen for MIDI data on?

1: Software In
•2: Software Out
3: Real Time Sequencer
4: Java Sound Synthesizer

This should be set to your software's output port.
Use the arrow keys and enter to select.

Unlicensed v Beta 1.052

Again, this screen lists all of the MIDI devices on your system. You should now choose which device you want Katapult to listen on, for incoming MIDI data. This is the device that your software will send MIDI data to, in order to update the Launchpad surface. Select the appropriate input device, and press enter. Again, if your software doesn't publish it's own MIDI devices, you will have to create a virtual device.

***Tip!** The Katapult input device is the device that Katapult listens on for incoming MIDI data. This should be set to the device your software uses to send MIDI data to, that are meant to update the Launchpad surface.*

The MIDI data flows like this:

Software -> Input Device (in this example named "Software Out" -> Katapult

If you don't want to use an input device, you can just select a device that's not in use, for example the "Real Time Sequencer" (this is a Java MIDI device, that you most probably wont be needing).

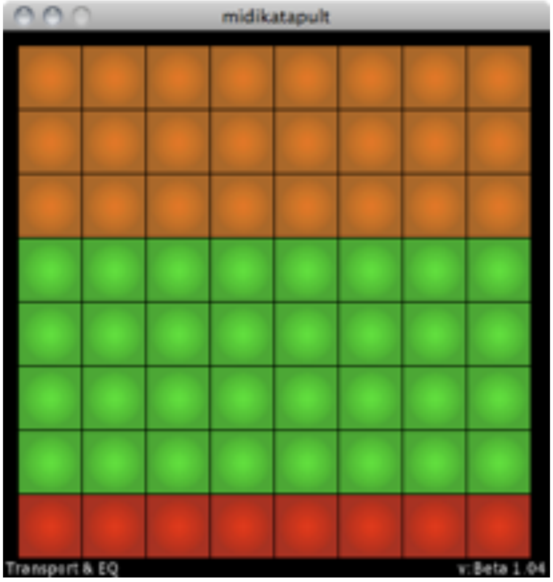
Be sure not to set the input device to the same device as the output device, unless you absolutely know what you are doing, and actually want this. Otherwise, Katapult will probably get very confused.

When you have selected both your input and output device, you will be presented with the following screen:

On this screen, you will have to choose whether you want Katapult to save the device choices you just made. If you choose “Yes”, Katapult will save your choices, and won’t ask you again when starting up. If you choose “No”, Katapult will ask you for devices again next time you open it.

Tip! *If you choose to save your device choices, and want to change them at a later time, open up the “config.txt” file, and delete the two lines starting with “in=” and “out=”. You can also just manually type in the names of the devices you want to use in the “config.txt” file.*

When you have chosen whether to save your devices or not, Katapult will load it's default layouts. If you're using an iPad, Katapult will display "Waiting for network connerction" until the iPad connects. When a connection has been established, you will be presented with this screen:



This screen will show you a representation of the current state and layout of the Launchpad or iPad, along with the name of the page you are currently on (lower left corner). You can now flip between pages using the "page" buttons on the Launchpad, or bring up a page selector by hitting the lower right button on the Launchpad (labeled "arm", glowing orange).

You can now open up your music software, and use it's "learning" function to assign the controls on the Launchpad to controls in your software. If your software doesn't have a "learn" function, refer to the "MIDI Output Format" section, for information on how to manually assign controls.

Tip! *If you hit the "Mixer" button on the Launchpad, you will be treated by a little light-show. Press the "Mixer" button again to turn off.*

When you have gone through this guide, please see the "Creating Layouts" section for information on how to create your own layouts.

If you are using Traktor Pro, you can load the "Sample.tsi" file from the "Samples" folder into Traktor. This is an example mapping for Traktor, corresponding to the first page of Katapults sample layout.

Creating Layouts

The basics

This section will explain how to create your own layouts. Katapult Layouts are created by writing a plain text file with short commands on separate lines for creating the individual controls. Here is an example:

```
page 1 My first page!
button 0 0 toggle
button 1 0 hold
xfader 0 1 8 takeover=3

page 2 Another page
pad 0 0 8 8 takeover=2
```

The first line tells Katapult that we are starting the first page of our layout. We also want to give this page a name, so we write “My first page!” after the page number. On the next line we create a toggle button at x=0, y=0 in the grid (this is the upper left corner of the Launchpad). After that line, we create a hold button right next to the toggle button. We also want a horizontal fader, so we create that at x=0, y=0 with a size of 8. We also sot the takeover of this fader to 3 milliseconds by adding “takeover=3” at the end of the line. Using takeover on faders and pads makes them move smoothly from one value to another. A low takeover number will make the control move quickly, and a high takeover value will make the control move slowly.

We then add another page by entering “page 2”, and again we add a page name after the page number. You don’t have to specify page names, but it can be useful if you have many pages in your layout. On this page we want to create one big 2D pad, so we enter “pad 0 0 8 8”, creating the pad at x=0 and y=0, with a vertical and horizontal size of 8, filling up the entire Launchpad surface. We also set the takeover to 2 milliseconds by adding “takeover=2” in the end.

That’s the basic concepts for creating your own layouts. Please open the “layouts.txt” file to see an example of a larger layout.

Control types

This is a list of all of the different control types you can use in Katapult.

Hold button

Created with : button X Y hold [persistent] [velocity=X]
Example : button 3 3 hold persistent velocity=80
Description : Hold buttons sends an on-signal (MIDI value 127) when they're pressed down, and an off-signal (MIDI value 0) when they are released. If persistent is set, the button will only update its color from external software. If the velocity argument is given, a custom velocity can be set.

Toggle button

Created with : button X Y toggle [persistent] [velocity=X]
Example : button 0 7 toggle velocity=115
Description : Toggle buttons toggle between sending on and off signals when they are pressed.

CC

Created with : cc X Y CC VALUE
Example : cc 0 0 3 115
Description : Creates a button, that will transmit a specific MIDI Controller Change message when pressed.

Program Change

Created with : pc X Y PROGRAM
Example : pc 0 0 3
Description : Creates a button, that will transmit a MIDI Program Change message when pressed.

Note

Created with : note X Y OCTAVE NOTE VELOCITY [toggle]
Example : note 0 0 3 c# 115
Description : Creates a button, that will transmit a specific MIDI note when pressed. If the toggle argument is given, the note-off event will not be sent until the control is pressed again.

Raw Note

Created with : rawnote X Y MIDINOTENUMBER VELOCITY [toggle]
Example : rawnote 0 0 87 115
Description : Creates a button, that will transmit a specific MIDI note when pressed. In creating a raw note, specify the MIDI note value to be transmitted (0-127).

Horizontal fader

Created with : xfader X Y SIZE
Example : xfader 0 0 8
Description : Faders create a line of segments that can be turned up or down by sliding your finger across them or by pressing in any location along them. Faders send MIDI value 127 when turned all the way up, and midi value 0 when turned all the way down, and intermittent values when set anywhere in-between.

Vertical fader

Created with : yfader X Y SIZE
Example : yfader 0 7 8

Description : A vertical fader works in exactly the same way as a horizontal fader, except that it’s vertical.

Inverted Horizontal fader

Created with : ixfader X Y SIZE
Example : ixfader 0 0 8
Description : Same as a normal horizontal fader, except that it’s inverted.

Inverted Vertical fader

Created with : iyfader X Y SIZE
Example : iyfader 0 7 8
Description : Same as a normal vertical fader, except that it’s inverted.

Horizontal slider

Created with : xslider X Y SIZE GRANULARITY
Example : xslider 0 0 8 5
Description : A slider works in the same way as a fader, but instead of setting a value directly by pressing a certain place on the fader, you slide your finger up or down the slider to increment/decrement its value. The GRANULARITY argument sets how much the slider is incremented/decremented with. In the above example, the slider value will be incremented with 5 for each slide of one segment.

Vertical slider

Created with : yslider X Y SIZE GRANULARITY
Example : yslider 0 7 8 5
Description : A vertical slider works in exactly the same way as a horizontal slider, except that it’s vertical.

Cross-fader

Created with : crsfader X Y SIZE
Example : crsfader 1 4 6
Description : A crossfader is a type of fader control that is incremented and decremented by pressing and holding on an area to the right or left of the center of the control.

2D Pad

Created with : pad X Y XSIZE YSIZE [invertx=yes] [inverty=yes]
Example : pad 2 2 4 4 invertx=yes
Description : Pads are like two-dimensional faders. They send a seperate MIDI value for each of their axes. The optional invert arguments can be specified to invert the output of MIDI data on each of the axis.

Drumrack

Created with : drumrack X Y XSIZE YSIZE STARTOCTAVE STARTNOTE VELOCITY [invert]
Example : drumrack 0 0 4 4 3 c 120 invert
Description : A drumrack creates a 2D grid of note-sending buttons, each one half-note apart, ideal for controlling drums. If the invert argument is given, the vertical ordering of the notes will be reversed.

Indicator LED

Created with : led X Y
Example : led 0 7
Description : Creates an indicator LED for displaying input from

software. LED controls doesn't send any MIDI out, but can be updated from chained controls or software.

Horizontal meter

Created with : xmeter X Y SIZE
Example : xmeter 0 1 8
Description : A meter works as a graphical display of for example the sync phase for a Track.

Vertical meter

Created with : ymeter X Y SIZE
Example : ymeter 0 7 8
Description : A vertical meter

Horizontal progress-bar

Created with : xprogress X Y SIZE
Example : xprogress 0 1 8
Description : A progress bar is identical to a fader, except that it doesn’t take input. It can be used for displaying track progress or volume levels, for example.

Vertical progress-bar

Created with : yprogress X Y SIZE
Example : yprogress 0 1 8
Description : Same as a horizontal progress-bar, except that it’s vertical.

Keyboard shortcut

Created with : kbd X Y KEYCOMBINATION
Example : kbd 0 0 control+alt+delete
Description : Creates a button that triggers a keyboard shortcut.

Other entries

New page

Created with : page NUMBER NAME
Example : page 1 A new page!
Description : This creates a new page in your layout. All controls must be mapped to a specific page. Start the layouts.txt file with the line "page 1". Define controls for this page beneath it. When done, add another page entry, "page 2", define more controls and so forth. Specifying a page name is optional.

Takeover

Created with : CONTROL takeover=X
Example : yfader 0 7 8 takeover=5
Description : Adds takeover to the control. Takeover can be used on faders and pads. The value after "takeover=" sets how quickly the control will move from one value to another. If the takeover value is set to 5, Katapult will wait 5 milliseconds between each step through the values, so if a control is moving from a value of 0 to a value of 127 (minimum to maximum) with a takeover value of 5, it will take a total of 5*127=635 milliseconds, or 0.6 seconds for the control to completely update. All of the intermittent values will be stepped through, so the movement in the software will be totally smooth.

Controller chaining

Created with : chain X1 Y1 to X2 Y2 [send]
Example : chain 0 0 to 4 0 send
Description : The above example chains the controller at 0,0 in the grid to the controller at 4,0 and instructs the chained controller to also transmit midi when it is updated. The send argument is optional, and if it is not set, only the internal state of the controller will be updated, but it won't transmit any MIDI, unless physically activated. Chaining is very useful when you want several different controls to influence the same parameter in a piece of software for example. This will ensure that all the controls on the Launchpad are in sync, so you wont get any erratic "jumps".

Color specification

Created with : oncolor [color]
offcolor [color]
defaultcolors
Example : oncolor green
offcolor redlow
Description : The above example sets the on-color (eg, when a button is pressed) to green, and the off-color (eg, when the button is inactive) to dim red. These colors will be used for all controls after they are specified, until new colors are specified, or the colors are set back to their default values by using "defaultcolors". If default colors are specified in the configuration file, these will be used. If not, Katapults standard default colors will be used. See the "Colors" section for a list of usable colors.

Custom channel specification

Created with : channel X
Example : channel 16
Description : This specifies that all controls after the “channel” line will send on channel X. To return to default channel assignments, enter “defaultchannels” on a separate line. All controls after that will then send on their default channels (page number). Channel number must be between 1 and 16.

Initial value specification

Created with : init X
Example : init 0
Description : This specifies that all controls after the “init” line will have their initial MIDI value set to X.

Colors

This is a list of the various colors that can be used in the layouts or configuration file:

- black
- red
- redlow
- green
- greenlow
- amber
- amberlow
- yellow

Additionally, these colors are available on the iPad:

- blue
- bluelow
- cyan
- cyanlow
- magenta
- magentalow
- white
- whitelow

Keys

This is a list of the various keys that can be used when creating keyboard shortcuts:

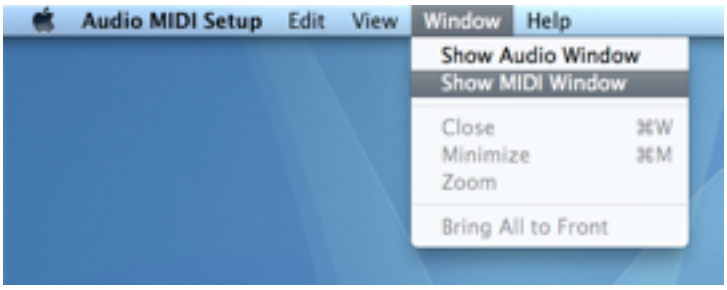
- 0 ... 9
- a ... z
- .
- ,
- +
- -
- f1 ... f19
- command
- control
- alt
- altgr
- shift
- space
- backspace
- delete
- up
- down
- left
- right
- tab
- esc

MIDI Devices

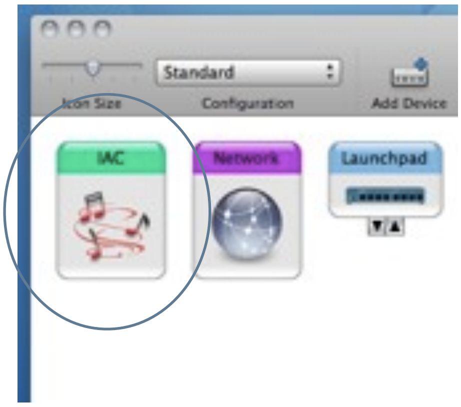
This section will explain how to create virtual MIDI devices if your software doesn't publish it's own MIDI devices. It's easy to create virtual MIDI devices, both on Mac and Windows.

For Mac

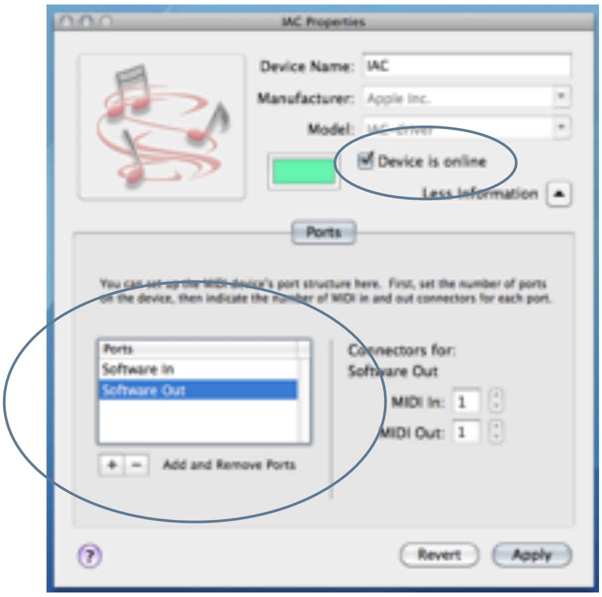
Go to your "Applications" folder, then the "Utilities" folder. Open "Audio MIDI Setup". In the "Audio MIDI Setup" application, make sure that "MIDI Studio" is visible:



In the "MIDI Studio" window, doubleclick the IAC icon:



In the "IAC Properties" window, make sure that "Device is online" is checked. Use the "+" button in the lower left part of the window to create two new ports, and name them as you choose:



IAC Properties

Device Name: IAC

Manufacturer: Apple Inc.

Model: IAC Driver

☑ Device is online

You can set up the MIDI device's port structure here. First, set the number of ports on the device, then indicate the number of MIDI in and out connectors for each port.

Ports
Software In
Software Out

Connectors for:
Software Out

MIDI In: 1

MIDI Out: 1

That's it! You have now created two virtual MIDI devices you can use with Katapult.

For Windows

I previously recommended using midiYoke for Windows users, but some users have seemed to have had problems getting the bi-directional communication working with that. I therefore suggest you use loopBe1 instead, as this seems to work much better. LoopBe1 can be found here: http://nerds.de/en/loopbe1.html.

I have kept the information on midiYoke below for reference.

There are various programs you can use for creating virtual MIDI devices. I suggest using midiYoke, as it is the easiest to use and set up. Go to the midiYoke webpage at: http://www.midiox.com/myoke.htm

Download midiYoke and install it. After the installation, restart your computer.

That’s it! You will now have several virtual MIDI devices you can use with Katapult (They’re named “Out to midiYoke: 1”, “Out to midiYoke: 2” and so on).

Bi-directional communication

Katapult can handle bi-directional communication, meaning that, as an example, when you change a virtual fader in a piece of software, the corresponding fader on the Launchpad surface is also updated. To use this feature, you must assign MIDI output from your software using the following format: "MIDI channel=Page, CC=(Y*8+X)". So, the MIDI channel your software outputs to determines what page the control is located on, and the CC number corresponds to the position in the grid, where the control is located.

MIDI Output Format

This is a brief description of how Katapult transmits MIDI. Normally, you should be able to use the "learn" function of your software to assign Katapult actions to actions in the software. If this is not possible, you can use this reference to manually set the values in your software. Katapult transmits MIDI data to the device you select when you start the program (Screen shows: "Select output device"). When a controller is changed on the Launchpad surface, Katapult transmits a MIDI Controller Change message, in the format of CC= (Y*8 + X), Value= (controller value). This message is sent to Channel= (page number). So if you press a button control on the surface, located at the first position (x=0) of the last row (y=7), on page 5 (channel=5), Katapult transmits CC56=127 to channel 5 of your selected MIDI device.

Note that pads send two seperate MIDI messages for each update. One on the normal output CC for the X axis, and one on CC+1 for the Y axis. So if you have a pad at x=0 y=3, it will send it's X value on CC24 and its Y value on CC25.

MIDI Input Format

The format for sending MIDI data to Katapult for updating the Launchpad controls is exactly the same as the output format. Just make sure you send MIDI data to the MIDI device you specified as Katapults input device. This means that if you use the learn function of your software to assign controls, take note of the CC and channel, and just set your software to transmit data to the same CC and channel on Katapults input device for updating that control.

Using Katapult alongside Ableton Live

If you want to use Katapult with Ableton Live, and preserve Lives session and clip launching controls, you can set up Katapult to only be active in one of the Launchpads usermodes. To do this you need to specify to lines in the “config.txt” file. Here is an example:

```
abletonlive=yes

liveusermode=1
```

Specifying these two options will mean, that pressing the “session” button on the Launchpad will deactivate Katapults control of the Launchpad, and delegate the control to Live. To reactivate your Katapult layout, press the “user 1” button on the Launchpad. The “liveusermode” setting can also be set to 2, which will make Katapult active when pressing the “user 2” button on the Launchpad.

Mapping the round side-buttons

The column of round buttons to the right of the grid can also be mapped. If you want to map a control to these buttons, specify it’s x position as 8. Note that only “button type” controls can be mapped to these buttons (note, rawnote, cc, button, pc, kbd).

Troubleshooting

Katapult crashes right after i start it!

Unfortunately, there is a bug in OS X's Java implementation, that makes the Java VM crash if a carbon-application is publishing it's own MIDI ports and Java tries to read from them. This means that Katapult will crash on startup, when it is scanning the available MIDI devices if a carbon application has published MIDI ports.

To avoid this, simply start Katapult before any carbon applications, and use IAC drivers instead of the applications own MIDI ports.

Also, there is the chance that you have LibMMJ installed on your system. After Apples latest Java updates, MMJ actually breaks Java's MIDI support instead of fixing it. Please try uninstalling LibMMJ from your system, and try again. The LibMMJ plugins can be located in any or all of these folders:

/Library/Java/Extensions
~/Library/Java/Extensions
/System/Library/Java/Extensions

Why wont Katapult start?

You need Java to run Katapult. If you are having trouble starting Katapult, please update your Java version.

If you are using Snow Leopard, make sure you have updated to at least 10.6.2, since there was a Java bug in prior versions, that will make Katapult cry. Also, make sure you have the latest Java updates from Apple. Go to "Software Update" and check if there is a Java update available. If you are still having trouble, send me an email, and I'll be happy to help you out!

Why can't Katapult detect my Launchpad?

Again, make sure that you have updated your Java runtime to the latest version. Also make sure that you have the latest Launchpad USB drivers installed. These can be downloaded from Novations website at: http://www.novationmusic.com/support/launchpad/

My layouts are not getting loaded properly!

Check to make sure that your "layouts.txt" file is written correctly. If you enter incorrect control parameters (such as placing a control out of the actual 8x8 grid), Katapult might stop loading your layout.

Using Katapult with Live

If you want to use Katapult with Ableton Live, you need to disable Live's Launchpad functions. Unfortunately Katapult and Live can't control the Launchpad at the same time, so you need to disable the Launchpad as a control surface in Live and use the IAC/midiYoke port as a normal MIDI input to Live instead.

I'm still having problems!

Feel free to contact me by email! I'll be happy to help you out :) You can also try and catch me on Skype.

Contact information

www.midikatapult.com

info@midikatapult.com

helpme@midikatapult.com

bugs@midikatapult.com

markqvist on Skype