© 2012 Luke Bradford

# Waveboard manual

Version 1.2

**Waveboard** is a tool for creating musical instruments.

The built-in instruments in Waveboard are good examples of how instruments are put together, and exploring them is a good way to learn how to make instruments. You can load an instrument at any time by tapping the **load instrument button**:

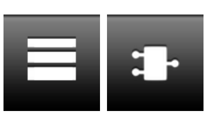
This manual will use an instrument called **Strawberry Pad** for its examples, so it might come in handy to load it and follow along.

## Basics

To build an instrument, you wire together different **components:** movable colored blocks with different functions. These range from input components like keyboards and touch pads to audio components like oscillators, envelopes, and speakers.

**Snap to note**, a typical component.

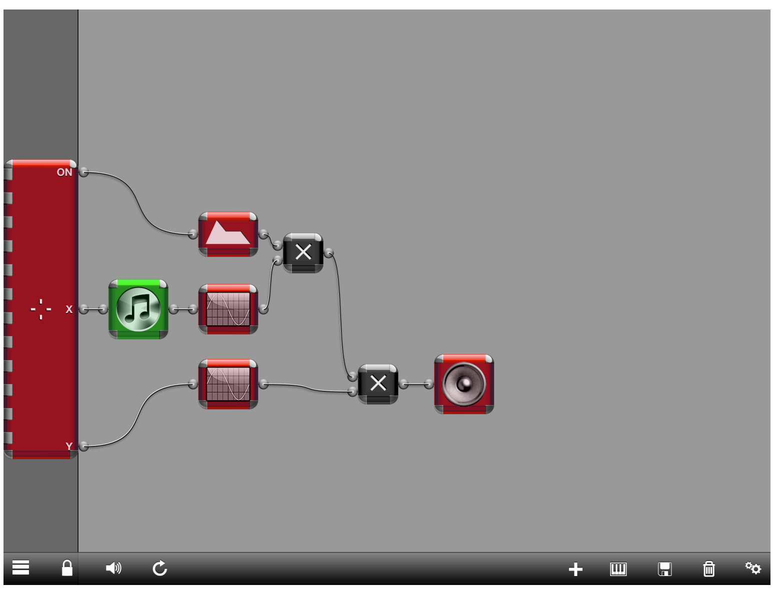
Each instrument has two main views, a **rack** and a **patch**. You can switch between them by pushing the rack/patch button at the leftmost side of the toolbar:



The rack view in Strawberry Pad has a single big red component – the touch pad. The patch view has several different small components that are wired together.

The rack view of Strawberry Pad.

You can think of the rack as the outside of the instrument. It holds large interface elements that are meant to be played, like touch pads and keyboards. These are called **rackable** components because they can be placed in the rack.

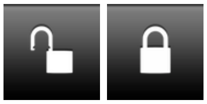



The patch view of Strawberry Pad.

The **patch** is the guts of the instrument. Most components – the ones that aren't 'playable' – are only in the patch view. So oscillators, speakers, and so on are just in the patch. Rackable components are also in the patch view – you can identify them by a tiled pattern on the left side of the component. To add rackable components to the rack, you snap them into the dark vertical stripe on the left side of the patch view, an area called the **mini-rack**. Components in the mini-rack will appear with the same height and vertical position in the rack. You can see this as you switch between the two views.

### Locking

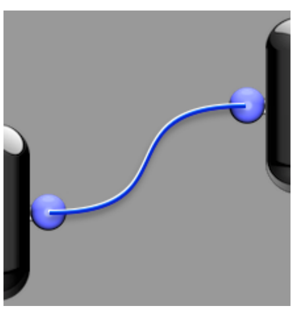
The whole instrument can be locked or unlocked using the **lock button**:



When the instrument is locked, its components and wires are frozen in place. You lock the instrument to play it, and unlock it to change the way it's built. By default, the rack view is always locked, so you can play it at all times. in Strawberry Pad, you can play the **touch pad** in the rack. Try playing it with one finger and see if you can hear what sliding left and right does and what sliding up and down does. Then try using more fingers. Components in Waveboard are **polyphonic** – they can each produce or process up to five voices at a time – so you can play the touch pad with five different touches at once.

When the instrument is unlocked, you can drag components around. In the rack, components can be dragged up and down, and in the patch they can be dragged anywhere. In the patch you can also use two-finger gestures to scroll and zoom. If the instrument is unlocked, you can also double tap any component to see its **detail view**. The detail view contains a description of the component and component-specific settings. In the detail view you can also delete or duplicate the component.

## Wires and signals

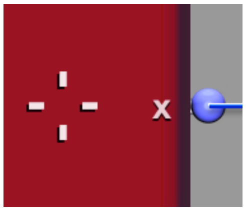


A wire connecting an outlet to an inlet.

When the instrument is unlocked, you can also rewire components in the patch. Components are connected by **wires** which carry a **signal** from one component to the next. They each run from one **outlet** on the right side of a component to an **inlet** on the left side of another component – these are the small metallic spheres that stick out of the sides of components. To create a wire, tap any outlet and then any inlet (or vice versa.) Signals move through the instrument, and each component changes the signal that passes through it.

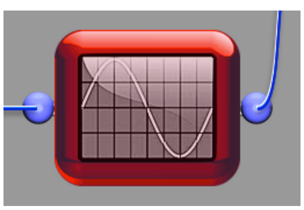
### Signal flow example

As an example of how signal flows through an instrument, let's track the path that signals take in Strawberry Pad.

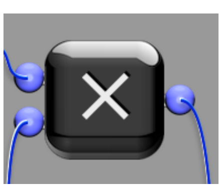



We'll start at the middle outlet of the touch pad, the one labeled 'x.' It produces a signal representing the horizontal position of the current touches. The range of these values is determined in the settings of the touch pad. So, for each touch, it sends a number representing the horizontal position of the touch. When the instrument is unlocked, you can double tap the touch pad to see its detail view, which includes the range of horizontal values – in this case, from 200.0 to 2000.0.

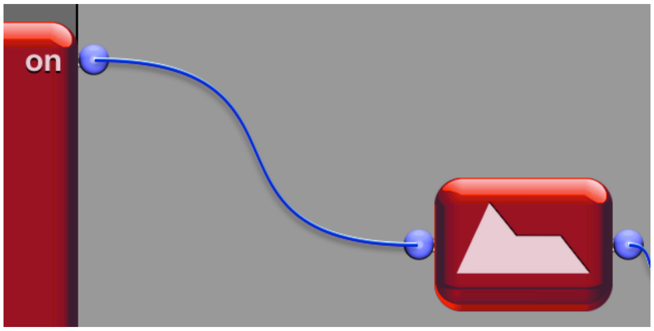
This signal goes into the **snap-to-note** component, which snaps its input value to the nearest note in a specified scale. In this case, it snaps to the F major pentatonic scale, as you can see in the snap- to-note component's settings. So, for example, if the component receives 443.61 as input, it will snap to 440.0 (A 440), which is the nearest note in the F major pentatonic scale. Frequencies in Waveboard are measured in Hertz (Hz), or cycles per second. The frequency of the note is the output from the component.



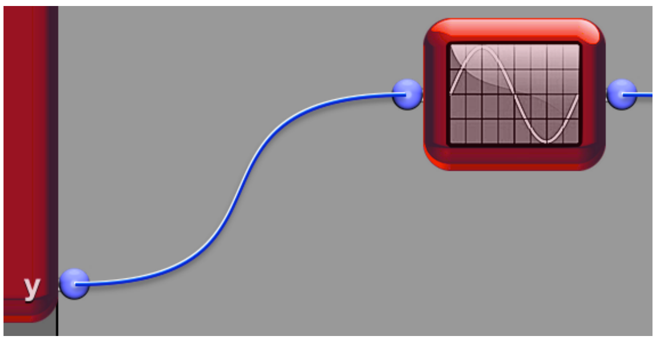
Next, the signal enters an **oscillator**. The oscillator takes the frequency provided by the snap-to-note component and generates a signal that repeats at that frequency. In this case it produces a pure sine tone, as the oscillator's settings show. All sound is made up of sine waves which oscillate at frequencies in the audible range. So this oscillator produces the sound that we'll hear.



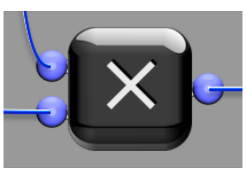
This sine tone signal enters a **multiplier**, which multiplies any two signals. The other input to the multiplier comes from the **ADSR component** above the oscillator, which has taken a different path.



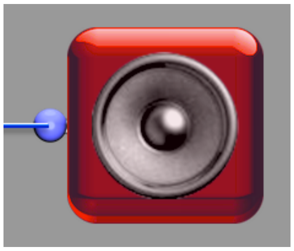

Back at the touch pad, an on/off signal starts at the top outlet, labeled 'on.' This signal just carries a 1 when a finger is touching the pad and a 0 otherwise. This signal enters the ADSR component, which turns the on/off signal into an attack-decay-sustain-release envelope. (Essentially, where the signal goes on and off suddenly, the ADSR envelope produces a gradual attack and release.) The signal from the ADSR component is multiplied by the signal from the first oscillator, as we saw above. The output from the multiplier enters another multiplier to be multiplied by yet another signal, this one from the lowest outlet of the touch pad.




This bottom signal begins at the outlet of the touch pad labeled 'y.' Like the X signal, it carries a value representing the Y position of the touches. The Y range of the touch pad is from 0.0 to 5.0, as you can see in the settings window. This signal enters another oscillator, which produces another sine wave. A sine wave between 0 and 5 Hz will not be audible, however. We're using this sine wave for a different purpose.



The signal from the oscillator is multiplied by the signal we explored above. This means that when the sine wave has a peak, the signal will be at its full volume, and when it has a valley, the signal will be silent. So this multiplication creates a heavy *tremolo*, the wobbling or wavering quality that you can hear when you play the instrument. The Y position of your touch controls the frequency of the tremolo.



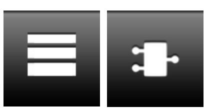
Finally, the output of the second multiplier enters a **speaker** component, which turns the signal we've been working with into actual sound from the iPad's speaker.

Strawberry Pad is a fairly simple, typical instrument. Other instruments use the same components in similar or different ways, so it's worth exploring a few instruments on your own and observing how they're put together.

## The toolbar

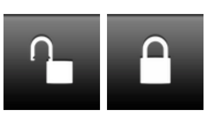
Waveboard's **toolbar** is located at the bottom of the interface at all times. You've already used a few of the buttons in the toolbar. Here's what each one does in detail.

### Rack/patch button



Tap this button to switch between the rack and the patch, the two main views of an instrument. The rack is the playable part of the instrument. The patch is where you wire together components to build the instrument itself.

### Lock button

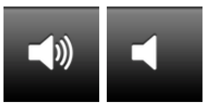


Tap this button to lock or unlock the instrument. When the instrument is unlocked, you can drag components around and connect them with wires. You can also double tap any component to see its detail view, a window with a description, component-specific settings, and buttons to delete or duplicate the component. In the patch view, you can also scroll and zoom with two-finger gestures.

When the instrument is locked, components and wires are frozen in place. Rackable components like touch pads and keyboards can be played when the instrument is locked. Other input components like switches, sliders, and number boxes also enable interaction when the instrument is locked. By default, the rack is always locked, so the lock button is disabled in the rack view. You can change this in the settings window.

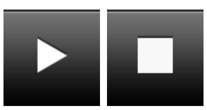
As a visual aid, components have metal caps on their corners and nodes and wires are gray when the instrument is locked. When it is unlocked, components do not have metal caps and nodes and wires are blue.

**Mute button**



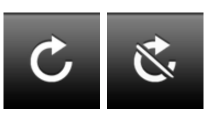
Tap this button to mute or unmute the instrument.

**Play/stop button**



Tap this button to start or stops all components in the instrument that have some notion of time. Currently only sequencers have start and stop functionality, so this button is only enabled when at least one sequencer is in the instrument.

**Rotation lock button**

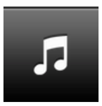


Tap this button to lock or unlock the autorotation of the interface. When unlocked, the interface will rotate to any orientation. When locked, it will never rotate.

**Add component button**

In the patch, tap this button to open the **add component** window. Then select any component from the list to add it to the instrument. A white glow will show you where the component will appear. Tap or drag anywhere to change its position. You can also double tap any empty spot in the patch to bring up the add component window.

**Song button**

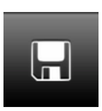


Tap this button to record, play, load, and save songs. You can play any instrument while a song is playing back. Songs are recorded as .wav files in mono at 44.1 kHz with 16-bit samples You can upload songs to your computer by accessing Waveboard’s documents folder in iTunes.

**Load instrument button**

Tap this button to open the **load instrument** window. You can load built-in instruments and instruments that you’ve saved.

**Save instrument button**



Tap this button to open the **save instrument** window, where you can save the current instrument with any name you want. Saved instruments can be loaded from the load instrument window. You can also upload them to your computer by accessing Waveboard’s documents folder in iTunes. Waveboard instruments have the file extension .wb.

### Erase instrument button

Tap this button to delete all components in the instrument and start with a clean slate.

### Settings button

Tap this button to open the **settings** window. There are various options to change the visual style and functionality of the interface.

## Under the hood

This section covers more advanced topics that require some knowledge of digital signal processing (DSP.)

### Control rate and signal rate

Waveboard runs at an audio sample rate or signal rate of 44.1 kHz. Components that do not deal directly with audio, including input components and sensors, run at a control rate of one sample per buffer. Buffers are normally 512 samples, so the control rate is about 86 Hz. Components that may or may not handle audio, such as math components, delays, and splitters, are equipped to handle any combination of control and signal rate signals. As a user, you can connect components without worrying about control rate and signal rate, and Waveboard will handle the appropriate conversions.

Waveboard uses 32-bit floating-point math for its audio processing.

### Pull

Waveboard uses a pull model to produce audio, so buffers are only requested from components that are upstream from at least one speaker in the signal chain.

### Polyphony

All components in Waveboard are polyphonic, and will produce or process five simultaneous voices. Rackable components can therefore handle five simultaneous touches. Math components will operate on all five voices, so empty voices may become apparent. For example, in the instrument Dancebot 1980, only one voice of the keyboard is audible. The keyboard’s frequency is multiplied by the sequencer’s value signal. Because there is only one cell on per step, the sequencer is producing one full voice and four empty voices. When multiplied by the keyboard's other four voices, these empty voices remain empty, so the next four voices are silent.

### Oscillators

For efficiency, oscillators in Waveboard use non-interpolated table-lookup synthesis or simple math to generate an oscillating signal. The sine table has 65,536 values. Square and sawtooth waves are not band-limited, so aliasing does occur.