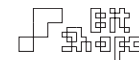


TC-11



TC-11

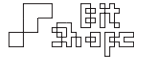
Version 1.7

User Guide

November 20, 2012

Kevin Schlei

Bit Shape Software



Contents

Getting Started

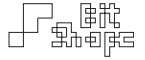
Introduction	4
TC-11 Overview	4
Interface Overview	5
1.1 Performance View	5
2.1 Navigation Bar.....	8
3.1 Load / Save View	9
4.1 Patch View	15
5.1 Settings View	18

Performance

Performance Basics	21
Voice Allocation.....	22
Device Motion	22
Connecting to an Amplifier	22
iPad Rotation	23
Performance Tips	23

Patch Editing

Controllers Explained	25
6.1 Voiced vs. Global Controllers.....	25



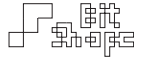
- 7.1 Controller Types26
- Synth Objects Explained30
- 8.1 Voiced vs. Mix Synth Objects31
- Programming Parameters33
- 9.1 Parameters Explained.....34
- 10.1 Parameter View.....35
- Module Programming.....41
- 11.1 AHDSR41
- 12.1 LFO.....43
- 13.1 Sequencer45

Recording

- Recording Capabilities52
- 14.1 Recording Controls52

Appendices

- Troubleshooting.....55
- 15.1 Issues / Solutions56
- 16.1 Single Touch Controllers.....57
- 16.2 Group Touch Controllers60
- 16.3 Device Motion Controllers62
- 16.4 Miscellaneous Controllers.....63
- Appendix B: Triggers.....63
- 17.1 Single Touch Triggers63
- 17.2 Group Touch Triggers.....64
- 17.3 Sequencer Triggers65
- 17.4 Miscellaneous Triggers65



Appendix C: Synth Objects66



Getting Started

Introduction

Thank you for purchasing **TC-11**. Your support is greatly appreciated, and it means I can continue developing innovative music apps for iOS and Mac OS X. **TC-11** is the product of countless hours of programming and research into synthesizer controller design.

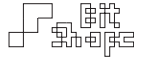
Your comments, feedback, and news are important to me. If you use **TC-11** in a performance or recording, please let me know through the website [www.bitshapesoftware.com] or email [bitshapesoftware@gmail.com]. Your creative output is one of the reasons making software instruments is rewarding.

TC-11 Overview

TC-11 is a programmable modular synthesizer on the iPad, controlled by multitouch and device motion controllers. All synthesis parameters can be controlled by these two sources, which allows the user to create countless unique patch configurations.

TC-11 does not have on-screen objects to control during performance. Instead, the performer's touches generate the control information. Some controllers are driven by a single touch. For example, the speed of the touch as it moves around the screen is its *Touch Speed*. Other controllers are generated with multiple touches, such as *Distance to Previous Touch*, where the positions of two touches are compared. Finally, the iPad's device motion capabilities can be used as controllers. The accelerometer, gyroscope and compass can be assigned to **Parameters** to turn your iPad into a expressive motion controller.¹

¹ Gyroscope and compass functionality is not available on the first generation iPad

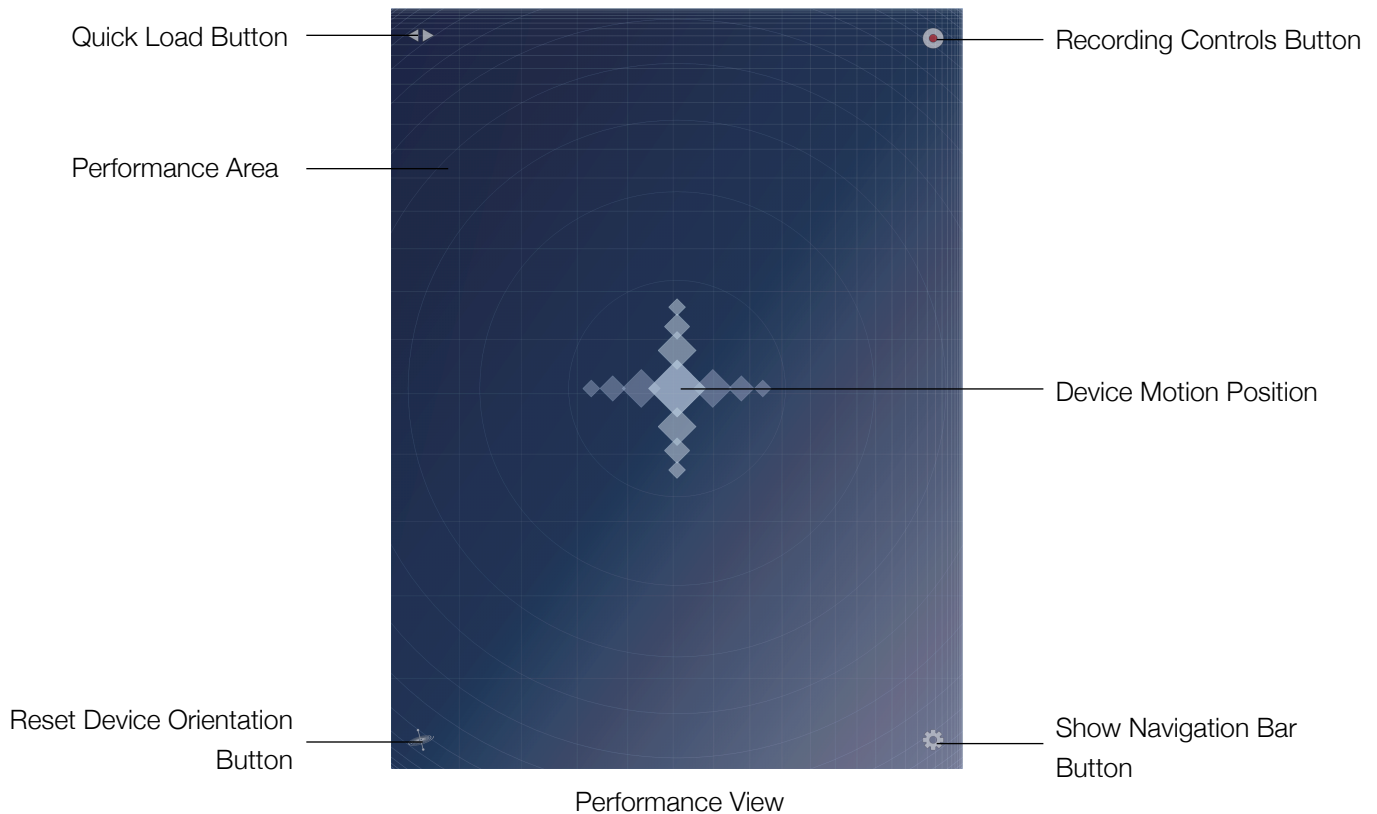


The synthesizer is modular, and uses common **Synth Objects** such as wavetable oscillators, filters, amplifiers, and effects. In addition, there are **AHDSR**, **LFO**, and **Sequencer Modules** that can be used for synth parameter control. Users can activate different **Synth Objects** and **Modules**, then assign multitouch controllers to them to build custom patches.

Interface Overview

In the following pages we will look at the individual sections of the **TC-11** interface. This will help you navigate through the app and familiarize you with the interface. More information about the individual objects and **Modules** can be found in later sections.

1.1 Performance View





1.2 Performance Area

The entire screen is used as the performance area. Use multiple simultaneous touches to control your patch's **Parameters**. As you touch the screen, the graphic display will update to show you which types of controllers are active, and what their positions or values are.

1.3 Device Motion Position

When device motion controllers are used (accelerometer, gyroscope, compass), the position of the device will be represented by shifting diamonds on the **Performance View**. Each axis of movement is represented by its own graphic. The accelerometer measures movement along the x-axis, y-axis, and z-axis of the device. The gyroscope measures the devices position as pitch, roll, and yaw. The compass points to the magnetic north heading. Move or rotate the device along its axes to determine the correct motion to use for each controller.

1.4 Reset Device Orientation Button

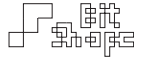
When the gyroscope is in use, you can reset the 'natural' position of the device by double-tapping this button. The pitch, roll, and yaw positions will all be zeroed. You can reset the orientation any time during performance.

1.5 Recording Controls Button

When this button is showing (it can be toggled to show / hide in the **Settings View**), double-tap it to reveal the recording controls. See [Recording, p. 52] for more information.

1.6 Show Navigation Bar Button

Double-tap this button to show the **TC-11 Navigation Bar**.



1.7 Quick Load Button

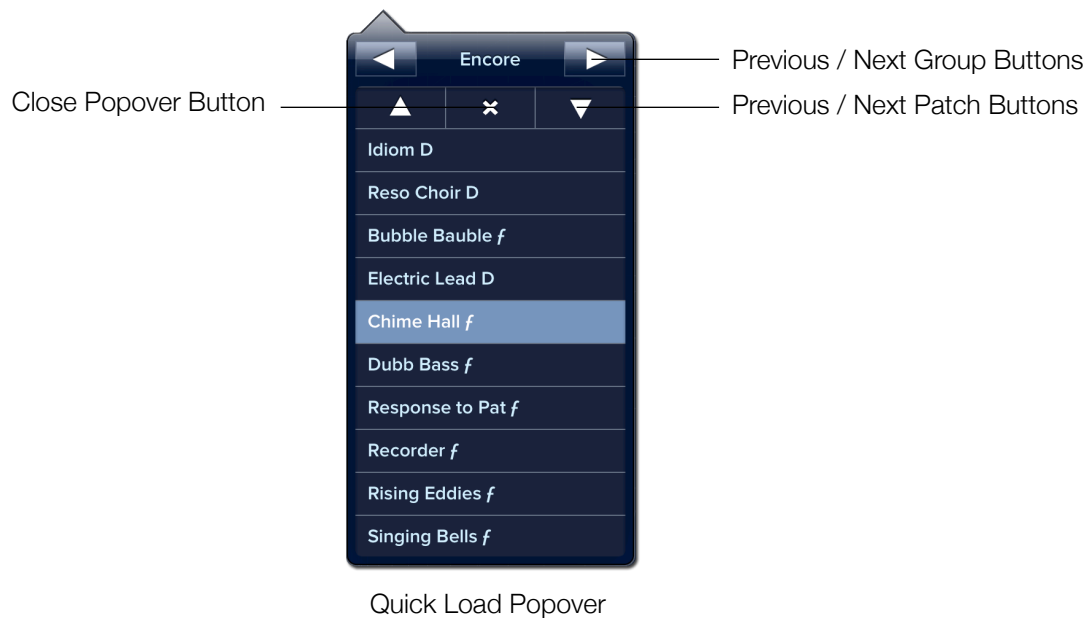
Double-tap this button to show a list of the current group of patches. The menu popover will stay open while you play. You can dismiss the popover by pressing the close button or any of the other buttons in the **Performance View**.

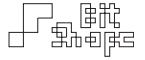
1.7 Quick Load Popover

The **Quick Load Popover** allows you to quickly switch to another patch during performance. It takes advantage of **Patch Groups**, which are collections of patches. You can create your own custom **Patch Groups** in the **Load / Save** view [Patch Groups, p. 9].

Pressing the previous / next group navigation buttons will cycle through all of your available groups. Choose a patch by tapping its name, or step from one patch to the next using the previous / next patch buttons.

To close the **Quick Load Popover**, tap the close popover button, the **Quick Load Button**, or any of the other buttons in the **Performance View**.





2.1 Navigation Bar



Navigation Bar

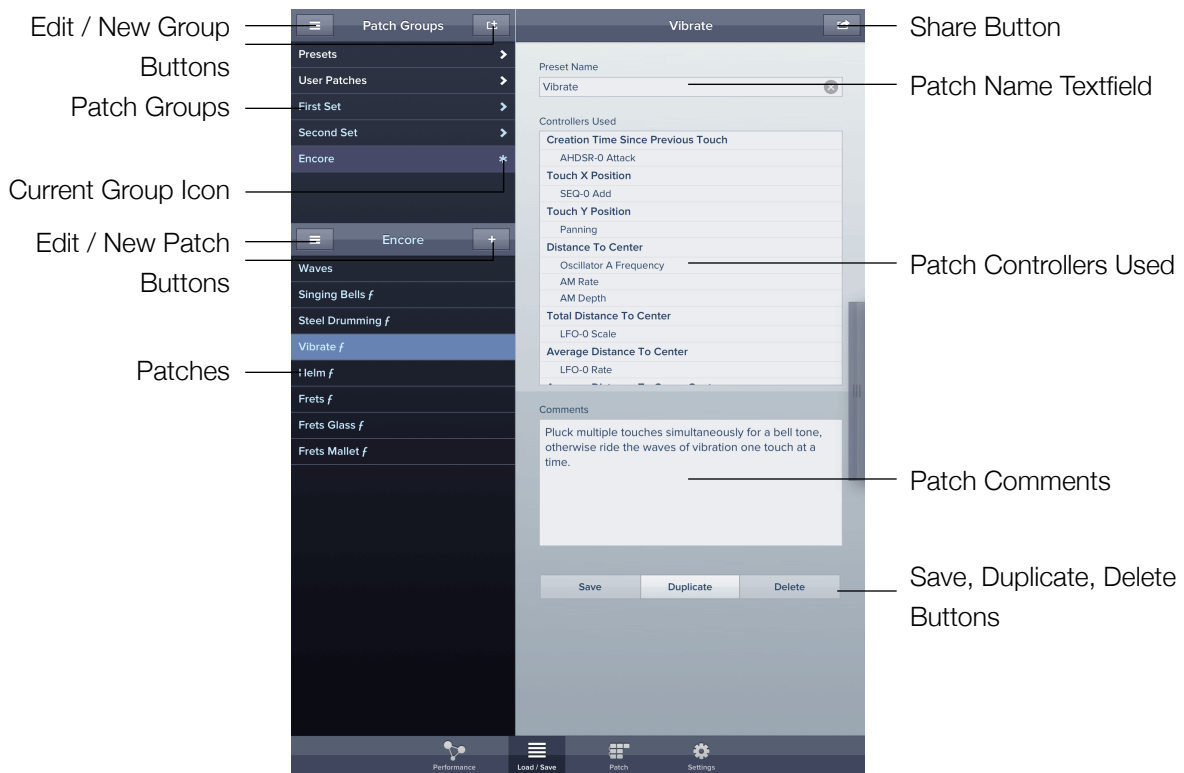
Show the **Navigation Bar** by double-tapping the **Show Navigation Bar Button** in the **Performance View**. Here you can switch between views. **TC-11** will fade out the sound when you switch away from the **Performance View**. The **Navigation Bar** will hide automatically when a touch is played in the **Performance View**.

2.2 TC-11 Views

The **Performance View** is where you can play the currently loaded patch. The **Load / Save View** is where you can load, save, duplicate, rename and share patches. In the **Patch View**, you can access the **Synth Objects** and **Modules** to edit the currently loaded patch. The **Settings View** allows you to change the display style and colors, and has a quick help reference area.



3.1 Load / Save View



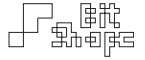
Load / Save View

The **Load / Save View** is your portal to loading, saving, duplicating, importing and exporting **TC-11** patches. It is also a place to organize and share your patches. This is the best place to start exploring what **TC-11** has to offer. The included presets demonstrate a variety of control styles and patch constructions.

When working on your own patches, **TC-11** will keep a working copy saved in a temporary location in case you leave the app. However, it is a good idea to save your work permanently by visiting this view.

3.2 Patch Groups

Patch Groups are collections of patches. There are two permanent groups: **Presets** and **User Patches**. The **Presets** group contains the **TC-11** preset patches. You cannot add or delete



patches from this group, but you can edit them and duplicate them into your own collection. The **User Patches** group is where your custom patches are stored. It will always show the full list of your patches. When new patches are created or imported, they will appear here. Duplicated preset patches will also be saved here.

You can create your own **Patch Group** to organize your patches. This is useful for performance, when you need a few specific patches to be ready to go at a moment's notice. It is also useful for sharing multiple patches without having to export them individually.

It is important to note that **Patch Groups** are really folders containing aliases (or shortcuts) to your patches. They are not duplicates. You can casually add or remove patches from groups without lasting consequences, but deleting a patch will permanently destroy the patch data.

Tip: Removing vs. Deleting Patches

*When you press the **Delete Patch Button** for a patch in a custom group, the view will change to highlight the patch in the **User Patches** group. This is a reminder that what you are actually deleting is the patch. If you simply wish to remove the patch from the group, swipe the patch name and press 'Remove,' or use the group organizer tables.*

Tap a patch name in this list to load one of the included **Preset Patches**. When the patch loads you will see its information displayed in the rightmost column. The included presets cannot be deleted or overwritten, but they can be duplicated into the **User Patches**. It is often a good idea to start with a preset patch, edit the parameters, and save your own customized version in the **User Patches**.

You can change the way patches are ordered in the **General** section of the **Settings** view. You can choose to show the patches in order (numbered, based on their creation date) or alphabetically.



3.3 New Patch Button

The **New Patch** button creates a blank patch template in your **User Patches**. Keep in mind that it may be easier to modify a pre-existing patch than start from scratch. Simply **Duplicate** the edited patch to save your new version.

3.4 User Patches

The **User Patches** list your saved custom patches. Tap the name of a patch to load it for performance. Patches that you build will be listed here along with any patches that you import via iTunes or email.

Tip: Backing Up Your Patches


Your **User Patches** will be listed in iTunes when you connect your iPad. Under the **Apps Tab** for your iPad in iTunes, scroll down to the **File Sharing** section. Here you can add or delete **TC-11** patches (even while the app is running), and copy them to your computer. It is advisable that you back up your patch collection frequently by copying the patch files to your desktop computer.

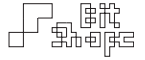
It is possible that patches created with newer versions of **TC-11** will not load in earlier versions. If that is the case, a warning will appear when you attempt to load it and the patch will be removed from your **TC-11** documents folder.

3.5 New Group Button

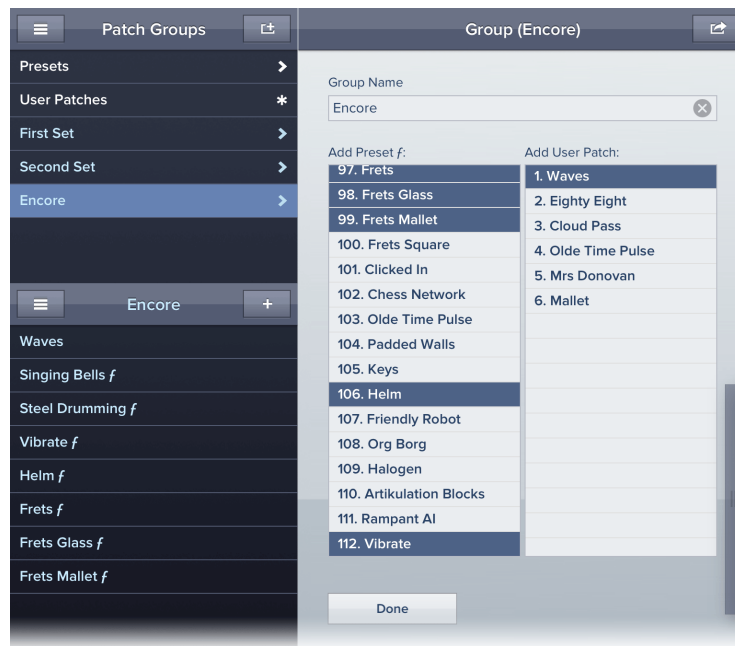
The **New Group** button creates a new **Patch Group** which you can fill with references to specific patches.

3.6 Organizing Patches Into Groups

Press the  button when a group is selected to bring up a list of available **Preset** and **User Patches**. Add and remove patches by tapping on the name of the patch.



Preset patches will appear in the group with a *f* symbol after their name to indicate that the patch is a preset. This allows you to distinguish between preset patches and user patches that may share the same name.



Group Organizer Tables

Once patches are in a group, you can rearrange the order by pressing the **Group Edit Button**, then dragging the patches by their handles. Later, in performance, you can step to the next / previous patch using the **Quick Load Popover**.



3.5 Share Button

Press this button to email the currently loaded patch or group to a friend. **TC-11** will automatically open a new email window with the patch or group included as an attachment. When the recipient receives the email on their iPad, they can simply tap the attachment to automatically load the file into **TC-11**.

Groups that contain preset patches will not duplicate the file data into the exported collection, but will retain the reference to the patch. The group should appear identical for other users who load it.

TC-11 patches have a '.tcp' file extension, and groups have a '.tcpg' extension.

3.6 Patch Name Textfield

Edit the currently loaded patch name. While **Preset Patches** cannot be overwritten, you can change the name then duplicate the patch into your **User Patches** for editing.

3.7 Patch Controllers Used

TC-11 has a large number of control options. When loading a patch, the **Controllers Used** list will show the controllers and **Modules** used by the patch, plus the **Parameters** they target. This list can help you understand how the patch can be played. It can also help pinpoint which synth parameter to change if you want to edit the patch.

3.8 Patch Comments

Saved with each patch is a **Comments** section. The **Comments** can include helpful information about the patch such as the author's name, specific performance instructions, or just general notes about the way the patch was built.

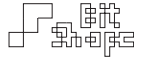


3.9 Save, Duplicate, and Delete Buttons

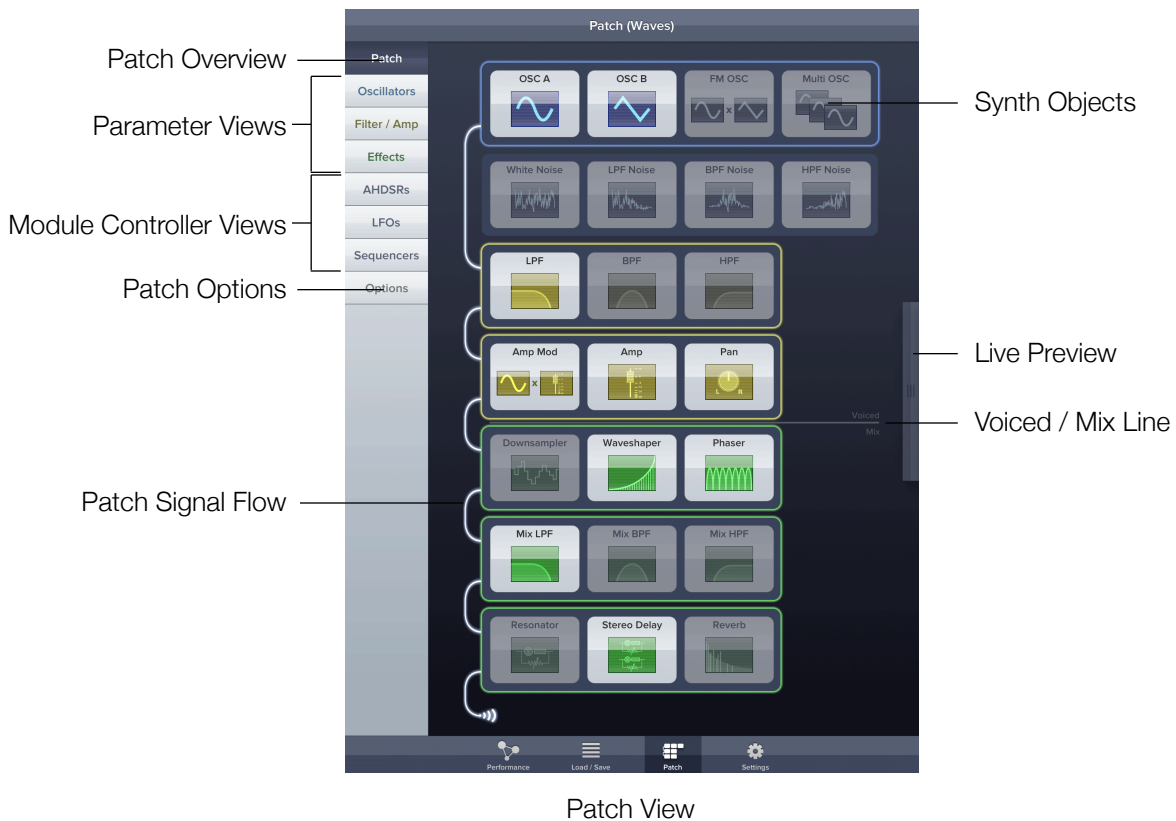
The **Save** button will overwrite the user patch with the name currently in the **Patch Name Textfield**. Because **Preset Patches** cannot be overwritten, this option is not available when the loaded patch is a preset.

The **Duplicate** button is always available. It makes a copy of the current patch and puts it in the **User Patches**. Use this button to 'save' **Preset Patches** into your **User Patches** for editing. If another user patch has the same name as the current patch, it will incrementally number the newly duplicated patch.

The **Delete** button permanently deletes the currently loaded patch. This action cannot be undone. Because **Preset Patches** cannot be overwritten, this option is not available when the loaded patch is a preset.



4.1 Patch View



The **Patch View** contains the guts of the **TC-11**, and always displays the currently loaded patch. It is the editing area. Pictured above is the **Patch Overview**, where you can enable the **Synth Objects** you want for your patch. By selecting a tab in the left column you can edit individual **Parameters**, or build a **Module** controller.

4.2 Patch Overview

Select this button to view the **Patch Overview**. Here you can see the **Synth Objects** listed from top to bottom. Tap an object once to enable or disable it. Groups of objects (objects sharing the same outline) may only allow one or two objects to be enabled simultaneously.



4.3 Parameter Views

Choosing the **Oscillator**, **Filter / Amp**, or **Effects** button will show the currently active **Parameters** for those groups. The content of these areas is determined by which **Synth Objects** are currently enabled. For example, when *Oscillator A* is enabled, the **Oscillator** tab will show the following **Parameters**: *Oscillator A Waveform*, *Oscillator A Frequency*, and *Oscillator A Level*.

More information about editing your patch in these views can be found in [Programming Parameters, p. 33].

4.4 Module Controller Views

TC-11 has three types of customizable **Module** controllers: envelope generators (**AHDSRs**), low frequency oscillators (**LFOs**), and step sequencers (**Sequencers**). Tap the corresponding button to create, edit or delete these controllers for your patch.

More information about working with **Modules** can be found in [Module Programming, p. 41].

4.5 Patch Signal Flow

The glowing line that connects the **Synth Object** groups shows the audio signal flow of your patch from top to bottom. For example, if the *Waveshaper* object is active, and the *Global LPF* object is active, the signal will pass through the *Waveshaper* before going through the *Global LPF*.

The speaker icon at the bottom is the end of the chain: your iPad's speakers or headphone output.



4.6 Synth Objects

The **Synth Objects** are the guts of your patch. By enabling objects you activate different parts of the synthesizer. Before diving into the deep control options, it is a good idea to understand the basic flow of your patch.

More information about patch construction can be found in [Synth Objects Explained, p. 30].

4.7 Voiced / Mix Line

This line separates the **Synth Objects** that represent individual control of touch voices, and those that affect the combined group audio signal. Above the **Voiced / Mix Line** are voiced objects, and below are mixed objects.

More information about patch construction can be found in [Patch Editing, p. 25].

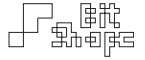
4.8 Live Preview

The **Live Preview** area is a way to hear your patches in real-time as you edit them. The drawer can be docked on the right edge of the screen to get out of the way, or positioned anywhere in the view. The view supports performance of each of the touch controllers, but it does not animate controllers.

While holding a touch on the view, the iPad will not auto-rotate to new orientations. This is helpful when auditioning **Device Motion Controllers**. If no touches are active in the **Live Preview**, the device will rotate normally.

4.9 Patch Options

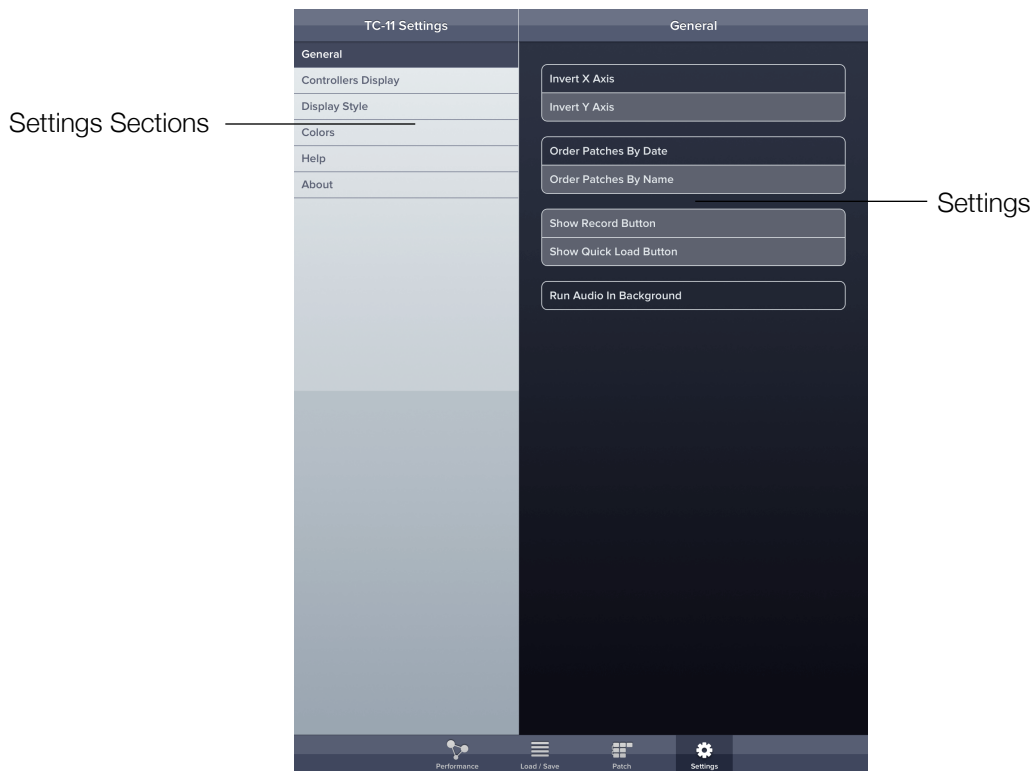
The **Patch Options** area contains settings that are tied to a specific patch. The **Suppress First Touch** setting mutes the voice that is triggered with the first touch of a group, while still allowing



that touch to generate controllers. This allows ‘stretchy’ patches to ignore the first touch, which is often used as a moving origin point for subsequent touches.

The **Custom Grid** options allow the patch to create a specific division of the x-axis, y-axis, and radial area of the performance view. The slope, number of divisions, and ‘checkered’ options change the look of the performance view, but do not actually affect controllers. They are meant as a guide for performance, often when using a **Sequencer** to generate a scale or division of the performance area.

5.1 Settings View



Settings View

The **Settings View** is where you can set global options that are not directly related to the current patch or synthesis, such as display settings and color changes. The **Settings View** also has a quick help reference section.



5.2 Settings Sections

The **General** section allows you to customize certain settings throughout the app. You can invert the screen axes, change the way patches are listed in the **Load / Save View**, and show or hide the **Show Record Controls Button** and **Quick Load Button** in the **Performance View**.

The **Display Style** section allows you to customize the way the controller data style. The option is between filled shapes and outlined shapes.

The **Controllers Display** section allows you to customize the way the **Performance View** displays controller data. You can invert the screen axes, and choose exactly which controllers you wish to display for aesthetic purposes.

The **Colors** section allows you to customize the color set for the **Performance View**. It is probably the most critical setting in the entire app for a stellar performance.

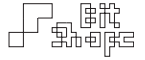
The **Help** and **About** sections contain information about **TC-11** and its author.

5.3 Invert Axes

One important aspect about **TC-11's Performance View** is that it does not rotate the performance area when the device is rotated. This means that the x and y-axes will stay locked with the physical screen of the device even if you rotate the **Navigation Bar** from vertical to horizontal.

Because each user may have a preferred way of holding their iPad, you can flip the positive / negative orientation of each axis separately. Controllers that directly reference the touch position on the screen will be affected.

It is important to realize that this option is for holding the device in a different way, and should not be used as a way to individually flip a particular controller. Each individual controller can be inverted, including axis-related controllers. If you have a stand or strap that holds your iPad in a particular way, it may be useful to set this permanently.

**Example:**

Load preset #7 - GrungeKick+Snare. The lower half of the x-axis is used for the kick drum, and the upper half is used for the snare. Then invert the x-axis in the Settings, and the roles are reversed without having to edit the patch.

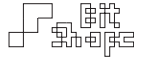
5.4 Override Controller Display

Each patch uses a set of controllers to drive its **Synth Objects**. By default, the only controllers that will draw to screen in the **Performance View** are those used by the currently loaded patch. This is to assist with performance: you can see exactly what touch movements matter, and which do not.

The **Override Patch Controller Display** option is for performers who want an unrelated set of controllers to display in the **Performance View** when they play. This is an aesthetic function only: controllers that have not been programmed into the patch will not affect **Parameters**.

5.5 Controllers For Display

If the **Override Patch Controller Display** option is turned on, you can choose which controllers to display in the **Performance View** from this list. Remember: this is an aesthetic option only. The controllers that are being displayed are not doing anything (except looking good).



Performance

Performance Basics

TC-11 was designed from the ground up as a multitouch driven synthesizer. That is to say, the multitouch data itself is what controls synthesis. Some synthesizers have buttons, sliders, and other on-screen objects that are controlled with a multitouch interface. In these systems, the multitouch capability is meant to enhance the interface accessibility, but it is not the means of control.

TC-11 takes a different approach. The multiple touches on the screen are examined, and relationships between them are used as controller data streams. Take this example: in a traditional synthesizer, there may be a knob that controls the synth volume. The user touches the screen to move the knob, which changes the volume. With **TC-11**, that job will be given to a touch controller. Perhaps the volume of a synth voice will be controlled by a touch's distance to another touch. The farther the distance between the touches, the louder the volume.

Example:

Load preset #3 - M Stretch. With two touches held down, move the touches closer and farther to one another and listen to the volume and modulation rate change. Then, twist the two touches around one another to change the angle, and hear the result.

The power of this system is the ability to create complex synth responses using sets of simple touch controllers. When a patch uses multiple touch controllers, the performative capabilities change dramatically depending on the style of performance. In the same way that a violin's sound changes based on many variables (bow position, bow speed, finger pressure, vibrato, etc.), **TC-11** patches can draw from many multitouch motions to create complex instruments.



Voice Allocation

Each touch in the **Performance View** is given its own synthesis voice allocation. In essence, each touch produces its own tone. When the touch begins, it is common for an **AHDSR** envelope generator to ‘open up’ the amplitude for that voice, much the same way a key on a keyboard starts a tone. However, it is not always so simple.

Some patches will allow the voices’ tones to be heard continuously, while the touches modify the way those tones are filtered. Others will delegate the starting and stopping of voice amplitude envelopes to a **Sequencer**. Regardless of how a voice is heard, the same one-to-one principle of touch-to-voice still applies.

TC-11 has eight [8] voice polyphony. Put another way, **TC-11** can support up to eight [8] simultaneous touches. When a ninth touch is placed, the earliest touch will be dropped in favor of the new touch.

Device Motion

The physical motion of the iPad can be used as a controller. You can turn and twist the device to change filters and effects, or even create vibrato through movement. The screen will display an orientation graphic that will move based on the motion of the device. Follow the graphic carefully to learn how to maneuver the iPad.

For more information about **Device Motion**, see [Device Motion Controllers, p. 27].

Connecting to an Amplifier

TC-11 generates full-frequency audio synthesis. While the iPad speaker is adequate for auditioning patches and casual performance, the full sound and stereo imaging of **TC-11** is best heard through a set of quality speakers. Connect the headphone jack output of the iPad to an



auxiliary input of your home stereo (using an 1/8" to stereo RCA cable), or to any professional mixing board and loudspeaker system for best results.

You may wish to disconnect the USB cable from the iPad, as unwanted bus noise can enter your signal output.

iPad Rotation

Since physically moving the iPad can be part of the synthesis generation, device rotation is disabled while in the **Performance** view. To temporarily enable rotation, double tap the **Show Navigation Bar Button**. You can then rotate the device to your current orientation.

The axes of the device are treated statically throughout **TC-11**. The rule is: the X-axis is always the left / right axis when holding the device with the Home button along the bottom, and the Y-axis is always up / down. There is a strong reason for this. When programming, it is significant which axis is assigned to a parameter. If the axes changed arbitrarily, it would be impossible to retain consistency in performance.

Performance Tips

When first performing any patch, try a variety of touch interactions. Some patches are designed to be held down for long durations, others are meant to be 'plucked' with short touches. Here are a few common gestures that can trigger complex responses from patches:

- Hold multiple touches down, and move them around the screen
- Change the number of touches on the screen
- Move touches around the screen quickly, or slowly
- 'Throw' the touches by moving them quickly, then releasing

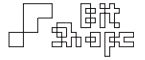


- ‘Pluck’ the screen by tapping and releasing quickly
- ‘Stretch’ a group of touches by moving them farther apart, then closer together
- ‘Twist’ a group of touches like you are turning a knob

Be aware that while some patches may feel as though they are sensitive to touch pressure, the iPad does not respond to touch pressure. Use a light touch when performing **TC-11**. The capacitive touch screen is sensitive to the charge of your finger, not the pressure it exerts on the screen (that is a resistive touch screen).

If you are performing **TC-11** in a live show, you may wish to turn off your iPad’s automatic sleep function (Auto-Lock). You can turn off this function in the iPad’s Settings, under General.

Also, be sure to turn off Multitasking Gestures in the iPad’s Settings, under General. Otherwise, moving groups of touches may close the app, or switch to another open app.



Patch Editing

Controllers Explained

Controllers are the way **TC-11** adjusts **Parameters** in real-time. They are the ‘invisible hands’ that change values like oscillator frequency, filter resonance, delay time, **LFO** rate, or any other value.

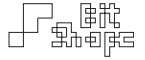
The majority of what makes a **TC-11** patch unique is its controller setup. A patch that uses a **Sequencer** to control the frequency of an oscillator will act very differently than one using the touches’ *Distance to First Touch*. When multiple controllers are used to control different areas of the patch, they generate exciting sonic results.

6.1 Voiced vs. Global Controllers

Some controllers generate their values by following a single touch voice [Voice Allocation, p. 22]. These controllers are ‘voiced’ controllers. A simple example of a voiced controller is *Touch X Position*. Each touch has its own position along the x-axis, and the controller reports that position *only* to the voice associated with that touch.

Other controllers may depend on the entire collection of touches to generate their value. These controllers are ‘global’ controllers. A basic global controller is *Group Count*. This controller sends a value based on the number of currently held touches (between 0-8). This number is not unique to a single touch voice, and so the controller will report the same value to *every* voice that follows it.

When assigning controllers to **Parameters**, it is important to know whether the parameter should follow a voiced or global controller. More information about controllers can be found in the following section.



7.1 Controller Types

TC-11 has a variety of controllers to use when building patches, most of which pull data directly from multitouch relationships. There are three types of controllers:

1. Touch-based controllers
2. **Device Motion** controllers
3. **Module** controllers (**LFO**, **AHDSR**, **Sequencer**)

In this section, the different controller types will be explained.

7.2 Touch-based Controllers

Touch-based controllers are created by touching the screen in the **Performance View**. Multitouch data streams are built by comparing these touches to one another.

There are two types of touch controllers. **Single Touch Controllers** are voiced controllers. ‘Single’ refers to the fact that the controller is following a single touch. In many cases more than one touch is needed to generate the data. For example, the *Angle To Previous Touch* checks the position of two touches to find the angle they create on the screen. The latter touch (the one created second) gets sent the *Angle To Previous Touch* controller data.

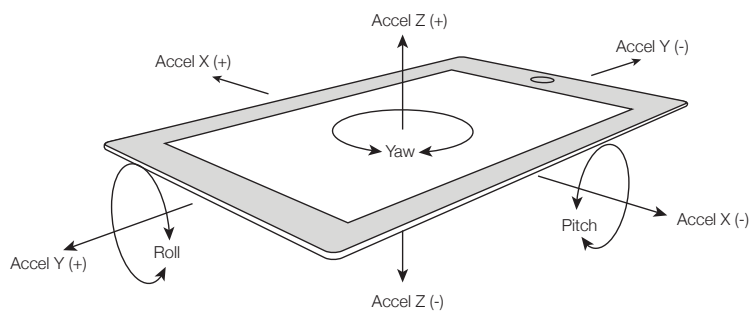
Group Touch Controllers are global controllers. They examine the entire collection of active touches (the group) to generate their data. One example is *Total Distance To Center*, which sums all of the touches’ distances to the center of the screen.

AHDSR and **Sequencers** need triggers to start and stop them. There are **Single Touch Triggers** and **Group Touch Triggers** that can target those parameters. **Single Touch Triggers** are voiced triggers, and **Group Touch Triggers** are global.



7.3 Device Motion Controllers

Device Motion controllers are global controllers that pull data from the physical movement of the iPad. There are three [3] device motion sensors: the accelerometer, the gyroscope, and the compass.² The accelerometer measures the rate of change in the device's motion through three [3] axes. The gyroscope measures the device's position in space by describing its rotation around three [3] axes. Finally, the compass compares the direction of the device to a fixed heading: magnetic north.



Accelerometer (x, y, z) and gyroscope
(pitch, roll, yaw) motions

TC-11 uses these data sources as continuous controllers. When the iPad's orientation is reset, gyroscope controllers will output their minimum value. As the device is rotated, controller values will increase *in either direction around their axis* until they reach the extremity of that movement.

Example:

Load preset #4 - Spyroid. This patch uses Gyroscope Roll to change the FM Depth. Reset the orientation (double-tap the Reset Orientation Button), then roll the device back and forth while holding a few touches to hear the change.

The accelerometer does not require an orientation reset. On the first generation iPad, it always references gravity. On later generations, it filters out gravity and only measures acceleration.

² Gyroscope and compass functionality is not available on the first generation iPad



The gyroscope can reset its natural position (orientation). Double-tap the **Reset Orientation Button** to zero the rotation values when you are in a comfortable resting position. The orientation is also reset automatically when you switch to the **Performance View**.

Since the compass reports the same motion as gyroscope yaw, it will not reset to a new fixed heading with the **Reset Orientation Button** is pressed. Use the compass when you want a permanently fixed spatial reference in your patch.

7.4 Module Controllers

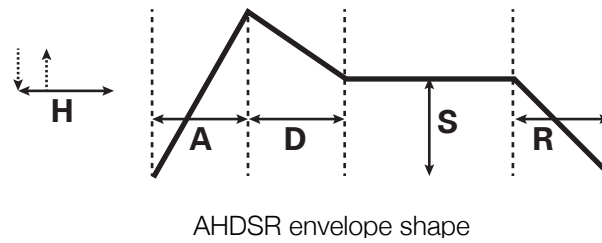
TC-11 has three [3] different **Module** controllers: **AHDSRs**, **LFOs**, and **Sequencers**. In this section their functionality will be explained. For a detailed look at programming **Modules**, see [Module Programming, p. 41].

7.4.1 AHDSR

AHDSRs are envelope generators. **AHDSR** stands for *attack, hold, decay, sustain, and release*. This **Module** controls a **Parameter** by moving its value smoothly along an envelope shape. It is commonly used to control the amplitude of individual voices. However, **AHDSRs** can be used to control any **Parameter**.

Each part of the **AHDSR** envelope shape can be changed in real-time by another controller. Plus, the output of the **AHDSR** can be scaled, allowing for complex shaping of a **Parameter**.

Hold is less common among envelope generators, and warrants explanation. A *Hold* time forces the envelope to continue even if the *Release* trigger has been activated. For example, if the *Hold* value is set to 500ms, and the user releases the envelope after 100ms, the envelope will continue through its path for another 400ms before actually releasing.



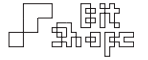
7.4.2 LFO

LFOs are low frequency oscillators. They are generally used to add a slow periodic change to a **Parameter**. Each **LFO** has an assignable waveform that cycles at controllable rate. In addition, the cycling waveform can be scaled (amplitude adjusted), and shifted (center position change). When a **Parameter** is controlled by an **LFO**, it will be modulated by the incoming slow-cycling waveform.

7.4.3 Sequencer

TC-11 has a powerful step **Sequencer**. The **Sequencer** stores up to 128 values which it can step through to generate a customized stream of values. All **Sequencer** parameters are controllable: playback rate, value adjustment, sequence length, and playback triggers. In addition, the **Sequencer** can ‘tune’ its values to note frequencies for certain **Parameters**, and even use a customizable note filter during playback.

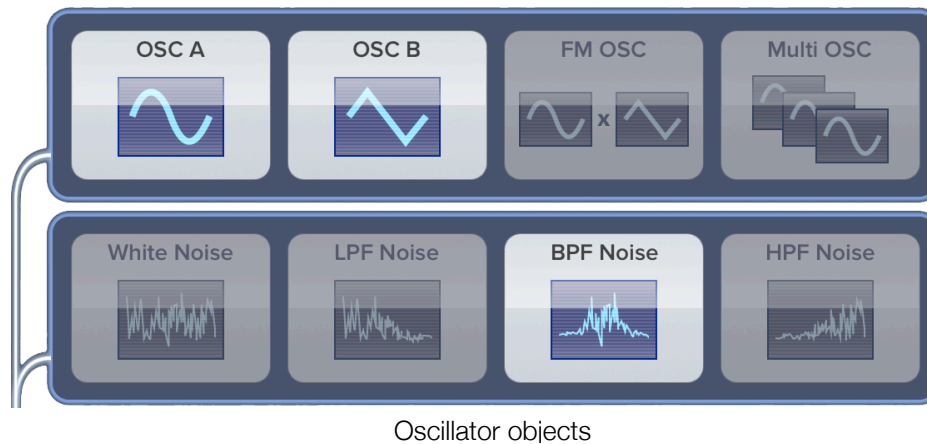
The **Sequencer** is the gateway to any traditionally pitched material, and its functionality is often used in creative patch construction. It is a complex tool which you can explore more fully in [Sequencer, p. 45].



Synth Objects Explained

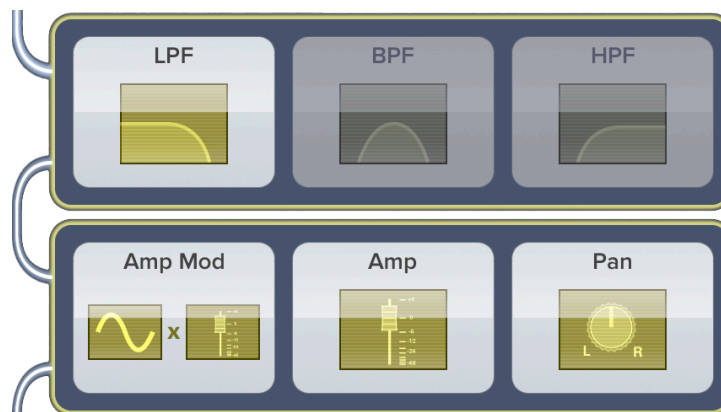
The **TC-11** synthesizer design is based on a set of modular **Synth Objects**. These objects generate or modify the sound in a patch. The number of objects and the connection order are both fixed, but the user decides which objects are active in a patch and which are not.

Switch to the **Patch View**, then to the **Patch Overview** to edit the **Synth Objects**.



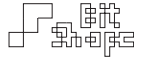
Oscillator objects

The blue **Synth Objects** make up the **Oscillator** group. There are wavetable oscillators: *Oscillator A*, *Oscillator B*, *FM Oscillator*, *Multi Oscillator*. Also available are filtered noise signals: *White Noise*, *LPF Noise* (low-pass filter), *BPF Noise* (band-pass filter), and *HPF Noise* (high-pass filter).

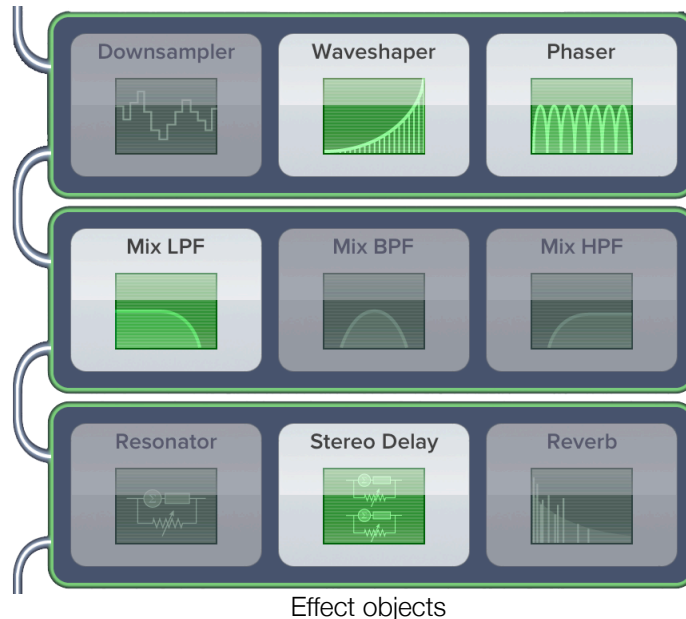


Filter / Amp objects

The yellow **Synth Objects** make up the **Filter / Amp** group. There are filters: *LPF* (low-pass filter), *BPF* (band-pass filter), and *HPF* (high-pass filter). Also included is an *Amp Mod* (amplitude



modulator). The *Amp* and *Pan* objects are always enabled, and control the amplitude and panning of the voice respectively.



The green **Synth Objects** are part of the **Effects** group. These include effects: *Downsampler*, *Waveshaper*, *Phaser*, *Resonator*, *Stereo Delay*, and *Reverb*. There is also a second group of filters.

Tap a **Synth Object** once to enable that object. If you double-tap the object, you will be taken to the **Parameters** area for that object's group. The **Parameters** corresponding to that object will be highlighted temporarily.

For a detailed description of each **Synth Object**, see [Appendix C: Synth Objects, p. 66].

8.1 Voiced vs. Mix Synth Objects

This section will explain the difference between **Voiced Synth Objects** and **Mix Synth Objects**.

8.2 Voiced Synth Objects

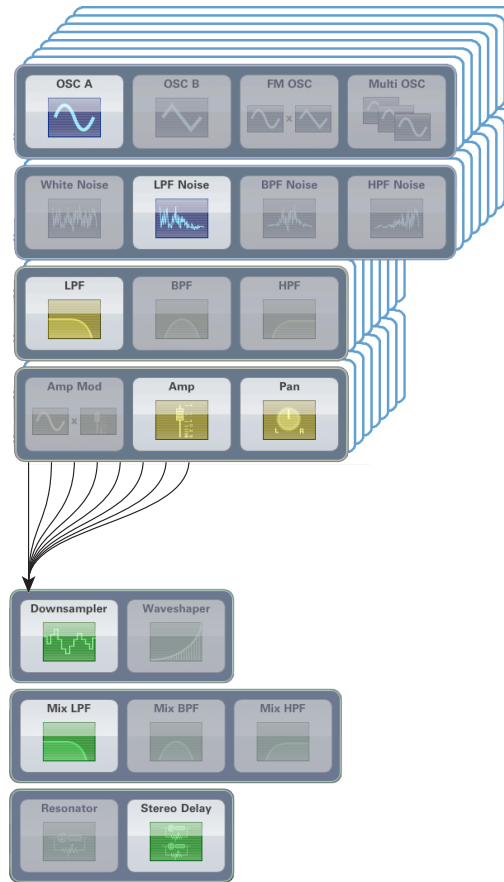
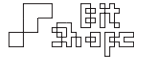


The **Synth Objects** above the **Voiced / Mix Line** are the objects that make up the eight [8] polyphonic voices. They are **Voiced Synth Objects**. Each touch voice is built with a copy of these objects, so they can be controlled separately from one another.

Keep this in mind when you are assigning controllers to these objects' **Parameters**. When editing objects in the **Oscillator** and **Filter / Amp** groups, you are setting how individual touches will be controlled. It is likely that you will want to use **Single Touch Controllers**, or **Module** controllers that are 'voiced.' Only choose a **Group Touch Controller** or 'global' **Module** controller if you want each voice to act exactly the same.

8.3 Mix Synth Objects

The **Synth Objects** below the **Voiced / Mix Line** (essentially the **Effects** group) are **Mix Synth Objects**. All eight [8] touch voices are mixed together before running through these objects. This means that **Effects** will act on the entire signal mix, not individual voices.

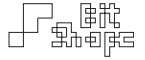


Voiced Synth Objects are mixed together before they run through Effects objects

Programming Parameters

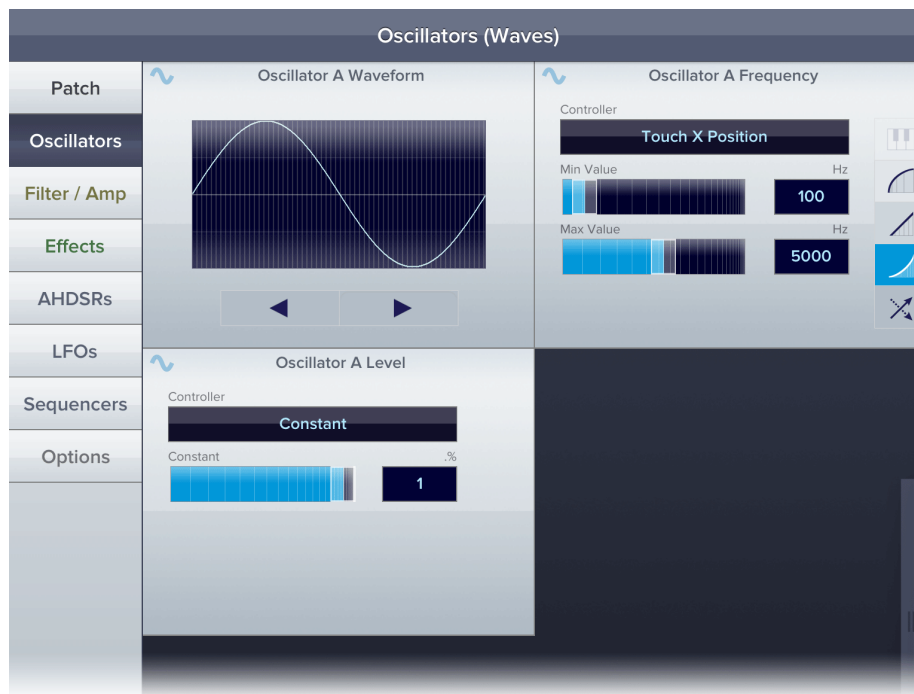
TC-11's uniqueness comes from the ability to control any synthesis parameter with any controller. Do you want to open a filter using a touch-based controller like *Group Total Speed*? Maybe you decide that tilting the device with *Gyroscope Roll* is more expressive. Or you could use an **LFO**, or **Sequencer**, or *Distance to Center*, etc., etc..

This section covers programming **Parameters** in detail.



9.1 Parameters Explained

Each **Synth Object** performs a specific synthesis function. The variables that control those functions are its **Parameters**. For example, *Oscillator A* is a wavetable oscillator that produces a tone. It has three [3] **Parameters**: *waveform*, *frequency*, and *level*. When *Oscillator A* is active in a patch, switching to the **Oscillators** view in the **Patch View** will show the following **Parameter Views**:

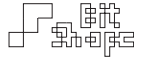


Oscillator A Parameters: *Oscillator A Waveform*, *Oscillator A Frequency*, and *Oscillator*

There are three [3] types of **Parameters**: general, waveform, and set-only.

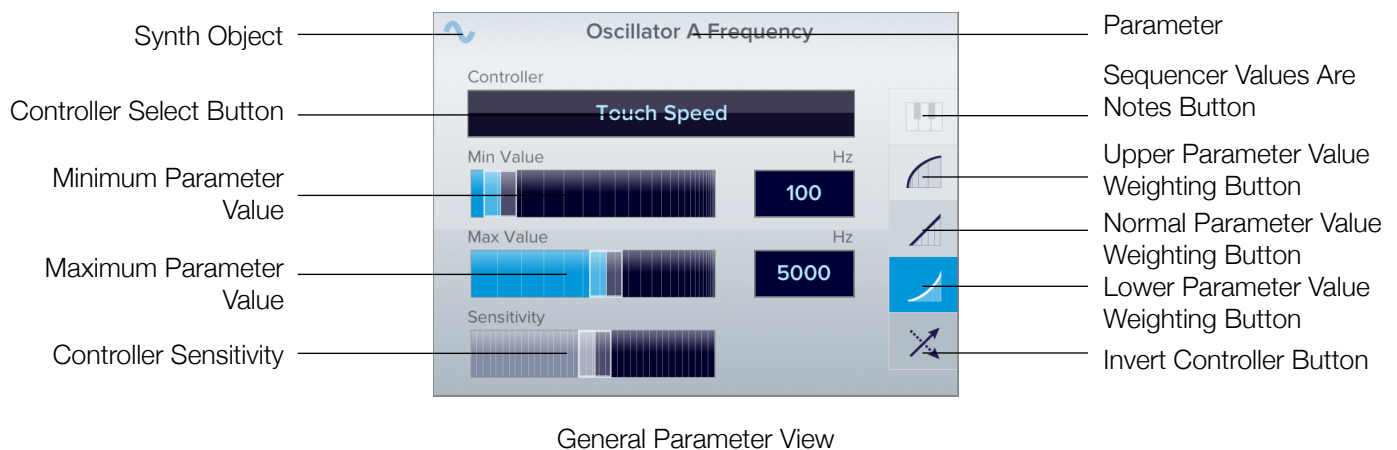
General Parameters are the most common. They have assignable controllers, variable value ranges, and can adjust the weighting of the incoming controller data. They can follow any controller type (touch-based, **Device Motion**, or **Module**).

Waveform Parameters are used when a **Synth Object** or **Module** has a selectable waveform. Touching the waveform opens a list of available waveforms. Touch any waveform to load it into the parameter. **Waveform Parameters** are static, so there are no controller options.



Set-Only Parameters are those that have a fixed value that cannot be changed during performance. One example is the *Multi-Oscillator Number of Oscillators* parameter in the **Oscillator** group. **Set-Only** parameters show a simple slider and textfield for static assignment.

10.1 Parameter View



10.2 Synth Object

This icon shows the **Synth Object** that contains the **Parameter**. **Synth Objects** may have multiple **Parameters**. Look for this icon to show you which **Parameters** are part of the same object.

Remember: activating / deactivating **Synth Objects** is done in the **Patch Overview**, see [Patch Overview, p. 15].

10.3 Parameter

A **Parameter** is a variable for a **Synth Object** or **Module**. The title at the top of the **Parameter View** indicates which **Parameter** the view is showing. For example, the **Parameter** shown above is *Oscillator A Frequency*. The name includes the **Synth Object** name (*Oscillator A*), along with the variable that is being targeted (*Frequency*).

One way to think of these **Parameter Views** is as **TC-11**'s replacement for a traditional knob or slider. Each view is there to change a single synthesis variable. Get to know what **Parameters**



are available, and you will be able to target the specific areas of synthesis that are important for your patch.

10.4 Controller Select Button

The **Controller Select Button** allows you to select which controller will target the **Parameter**. When you press the button, you will be shown a list of controller categories that are appropriate for that **Parameter**. You can navigate the menu to choose a controller. When you tap the controller name, it will be selected as the source for that **Parameter**. You can cancel the selection any time by tapping outside the menu popover.

This is where most of the fun happens in programming **TC-11**. Choosing a novel way of controlling a **Parameter** may open up a new sound or style of performance. Some controllers are driven very directly by the player, like touch-based controllers. Others, like the **LFO Module**, will run by themselves with as much (or as little) user interaction as you decide. When constructing a patch, deciding which areas of your patch you want to control directly is often a good first step.

All controllers have a minimum limit (start) and maximum limit (end). When the user generates values within this range, those values will be mapped to the **Parameter's** minimum to maximum values. The most basic example of this is the *Touch X Position* parameter. The far left of the screen is the controller's minimum limit (0 pixels), and the opposite side of the screen is the maximum limit (768 pixels). The position of the touch within that range would set the **Parameter's** value, mapped to whatever minimum / maximum value range the user has set.

For more information about controllers, see [Controllers Explained, p. 25].

10.5 Minimum / Maximum Parameter Values

Parameters have user-defined minimum and maximum values. This allows you to control the effective range of a parameter. For example, the *Oscillator A Frequency* **Parameter** might be set



to cover a range of 50Hz to 350Hz. This would set the oscillator to generate lower frequencies, which may be used for a bass patch.

The sliders allow for course setting of the minimum and maximum values. For fine control, you can type a value directly into the textfields next to the sliders.

Above the textfields is a label which shows the **Parameter's** unit of measurement. Common unit types are:

- Hz - Hertz, cycles per second
- BPM - beats per minute (beat = quarter note)
- .% - percentage, with a range of 0-1
- # - number, a direct correlate to the **Parameter**
- Q - quality factor, resonance

You cannot set a minimum parameter value higher than a maximum parameter value, and vice versa. If you would like the values to be inverted, press the **Invert Controller Button** to flip the incoming controller values.



10.6 Controller Sensitivity

Some controllers have a user-definable **Controller Sensitivity** setting. For example, you can decide whether *Touch Speed* reacts to slow movements in a drastic way (high sensitivity), or if it should require a violent gesture just to set it going (low sensitivity). Not only does this allow flexibility in performance, but it can dramatically change the sound of the patch.

Controller Sensitivity should be addressed by the user when using time-related controllers. For example, *Creation Time Since Previous Touch* can be used to simulate chording if the sensitivity is high enough.

Example:

Load preset #26 - Punch in the Face. Each time you create a touch it uses the time since the last touch to change the strength of the synth voice. Press multiple touches simultaneously for the big punch!

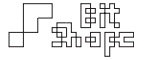
The **Controller Sensitivity** slider will appear automatically for those controllers that support it. Dragging the slider to the right makes the controller more sensitive; to the left makes it less sensitive.

10.7 Sequencer Values Are Notes Button

Some **Parameters**, generally those that deal with audio rate frequencies, can use a **Sequencer** to generate traditional note frequency values. When this button is active, the values from the **Sequencer** are no longer treated like a normal controller, but instead are considered MIDI note values.

To activate this feature, set the **Parameter** to be controlled by a **Sequencer**. If the **Sequencer Values Are Notes Button** becomes active and selectable, the **Parameter** allows for this type of control. If it does not, the **Parameter** does not support note values.

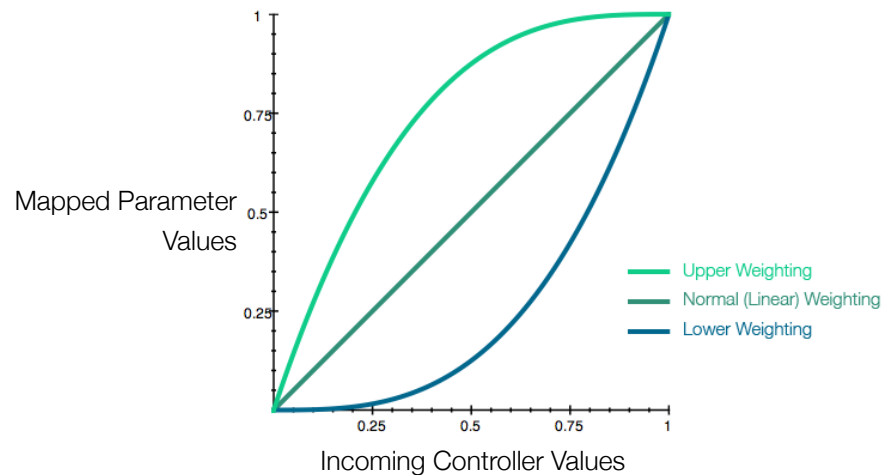
When the **Sequencer Values Are Notes Button** is activate, there are no controller or range values to set. Incoming **Sequencer** values are treated as MIDI note values (60 = middle C, 61 = C#, etc.). The values will also be filtered according to the **Sequencer's Key Filter**.



For more information about key filtering, see [Keyboard Filter Button, p. 46].

10.8 Parameter Weighting Buttons

Parameter weighting is when controller values are curved to favor a particular range of values when mapped. There are three different weightings: **Normal**, **Upper**, and **Lower Weighting**.



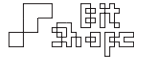
Parameter Weighting Comparison: Upper, Normal, and Lower Weightings

Normal Weighting is a one-to-one (linear) mapping of the controller range to the **Parameter** value range.

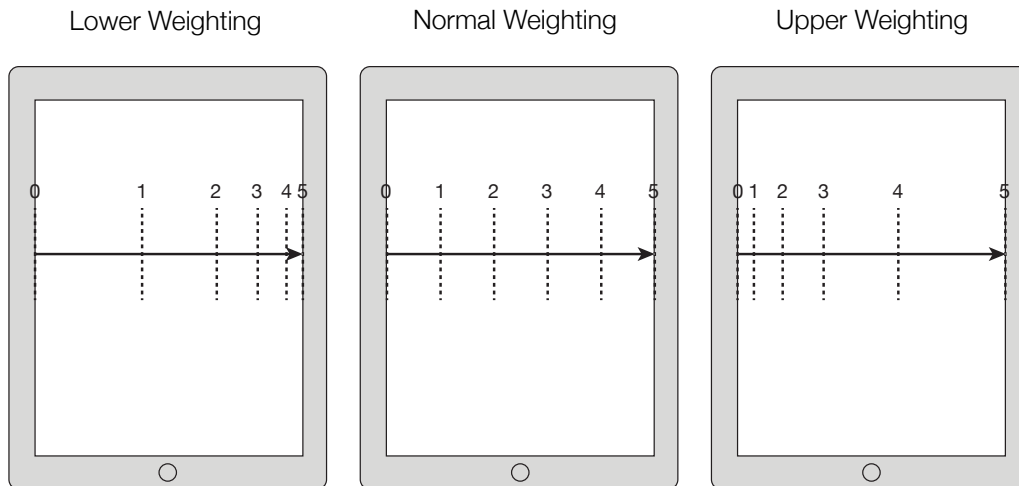
Upper Weighting pushes incoming controller values towards the upper range of the **Parameter** values. This means that incoming controller values will quickly push the **Parameter** values up towards their upper range.

Lower Weighting pulls incoming controller values towards the lower range of the **Parameter** values. This means that incoming controller values will keep **Parameter** values closer to the lower range of until they reach the very top.

Weighting is important for all parameters, but one clear example of its importance is when dealing with frequency. Take the following setup: *Oscillator A Frequency* is controlled by the



Touch X Position controller, and the min. / max. range is 50 Hz - 5000 Hz. If the weighting is set to **Normal Weighting**, the bottom few octaves (50 - 400 Hz) take up only 54 pixels of the screen. With **Upper Weighting**, it's only 28 pixels. But if the weighting is set to **Lower**, the same octaves use 204 pixels to sweep through, which acts much closer to the way we generally perceive frequencies.



Differences in weighting, with the *Touch X Position* mapped to a frequency parameter ranged 0-5 kHz

10.9 Invert Controller Button

The **Invert Controller Button** will 'flip' incoming controller values to reverse the mapping of the **Parameter** values. For example, if *Touch X Position* controls the frequency of an oscillator, the high frequencies will be towards the upper limit of the x-axis (where the grid lines get closer together in the **Performance View**). If the **Invert Controller Button** is activated, the high frequencies will be towards the lower limit of the x-axis instead.

This is particularly useful for time-based controllers. These controllers start at the lower value limit, and as time passes, move towards the upper limit. If you would like that direction to be reversed, use the **Invert Controller Button**.



Module Programming

The three [3] **Module** controllers in **TC-11** are the **AHDSR**, **LFO**, and **Sequencer**. In this section we will look at their functionality and some tips for using them in your patches.

11.1 AHDSR

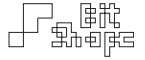
AHDSR View

The **AHDSR** is an envelope generator. When triggered, it ramps a control value along a user-defined envelope shape. Like traditional **AHDSRs**, the envelope starts at a zero [0] value, then ramps to its full value before returning to zero [0].

AHDSRs are commonly used to apply an amplitude envelope to individual voices. In **TC-11** you will almost always see the *Amp* object being controlled by a voiced **AHDSR**. **AHDSRs** can target any **Parameter** to shape its value.

11.2 Delete Button

The **Delete Button** will remove the **AHDSR** from the synthesis graph. Any **Parameters** that were controlled by the **AHDSR** will be reset to *Constant*.

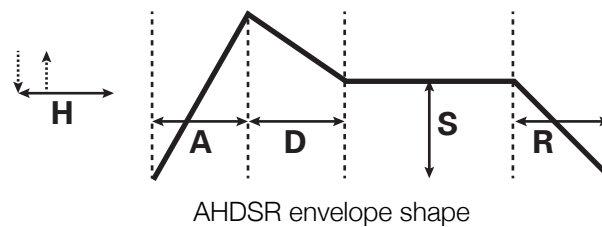


11.3 AHDSR Name

Each **AHDSR** will be assigned a unique name. This name will appear in the controller select popover when programming **Parameters**, and is saved with the patch.

11.4 AHDSR Shape

This image shows the shape of the **AHDSR** envelope.



11.5 AHDSR Trigger Parameters

There are two [2] trigger **Parameters** for each **AHDSR**: *Start* and *Release*. The *Start* trigger starts the envelope shape in motion. If the **AHDSR** is in the middle of an envelope, the *Start* trigger will reset the envelope to the beginning (zero [0]). The envelope will generate its values until it reaches the *Sustain Level*, where it will stay until the **AHDSR** receives a *Release* trigger. It then ramps to zero [0] over the *Release* time.

11.6 AHDSR Parameters

There are six [6] **Parameters** for each **AHDSR**: *Attack*, *Hold*, *Decay*, *Sustain*, *Release*, and *Scale*. *Attack*, *Decay*, and *Release* each sets the time (in milliseconds) that its portion of the envelope takes to complete. *Sustain* is the level of the envelope it will output indefinitely when it reaches that portion of the envelope.

Hold is less common among envelope generators, and warrants explanation. A *Hold* time forces the envelope to continue even if the *Release* trigger has been activated. For example, if the *Hold*

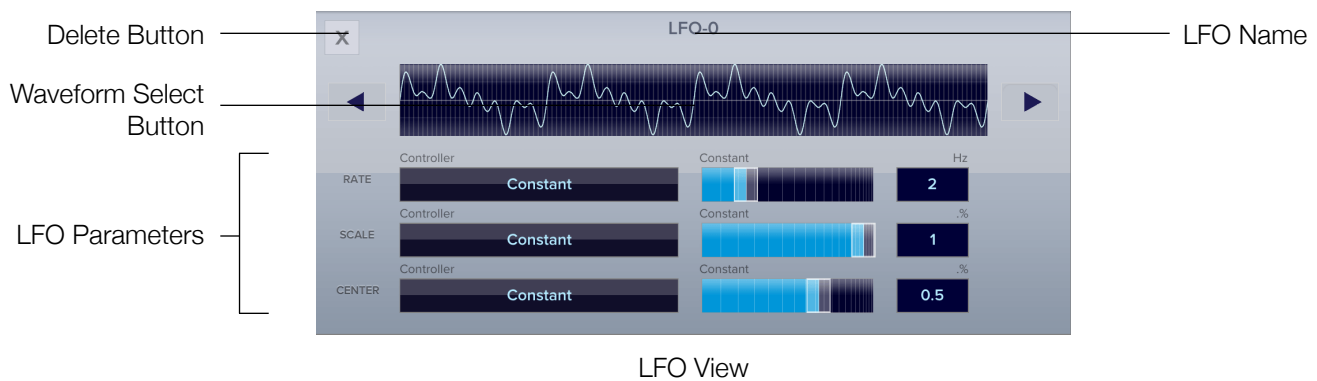


value is set to 500ms, and the user releases the envelope after 100ms, the envelope will continue through its path for another 400ms before actually releasing.

Scale allows the entire envelope shape to be stretched from full value (a Scale value of 1.0) to flat (a Scale value of 0.0).

While each **Parameter** can be controlled dynamically, note that the envelope will only set the value at the time of triggering. Once the envelope shape is in motion, changing the time of that **Parameter** will have no effect on the envelope until it has reached the next control point.

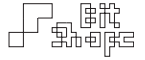
12.1 LFO



The **LFO** is a low frequency oscillator. It has a selectable waveform and basic control over the rate of oscillation and amplitude. **LFOs** are generally used for periodic control of **Parameters**, but can also facilitate quasi-random **Parameter** control.

12.2 Delete Button

The **Delete Button** will remove the **LFO** from the synthesis graph. Any **Parameters** that were controlled by the **LFO** will be reset to *Constant*.

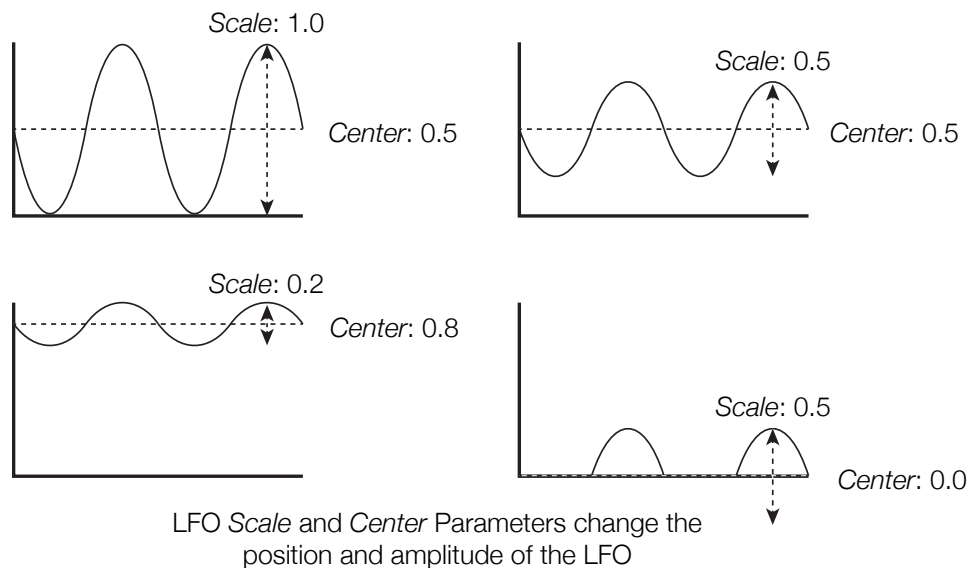


12.3 LFO Name

Each **LFO** will be assigned a unique name. This name will appear in the controller select popover when programming **Parameters**, and is saved with the patch.

12.4 LFO Waveform Select Button

This button shows the selectable waveforms for the **LFO**. Tap the image of a waveform in the waveform select popover to load it into the **LFO**.

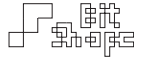


12.5 LFO Parameters

There are three [3] parameters that adjust the **LFO**: *Rate*, *Scale*, and *Center*. *Rate* sets the frequency of the waveform, and has a range of 0 Hz - 50 Hz. *Scale* adjusts the amplitude of the waveform. *Center* adjusts the mid-point of the waveform's amplitude.

Control values are scaled individually for each **Parameter** they control. If *Scale* is set to 100%, the **LFO** will cycle through the full range of the **Parameter**. When it is set to a lower percentage, it will only cycle through a portion of the **Parameter** range.

Center allows you to offset the **LFO** values within the defined **Parameter** range. When used in conjunction with *Scale*, the **LFO** can be a very flexible controller.



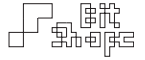
13.1 Sequencer

The image shows the Sequencer View interface with the following components labeled:

- Minimize View Button
- Rate Format Button
- Keyboard Filter Button
- Randomize Button
- Delete Button
- Sequencer Name (SEQ-0)
- Sequence Banks (1-16, 17-32, 33-48, 49-64, 65-80, 81-96, 97-112, 113-128)
- Sequence Graph
- Sequence Values (114, 20, 35, 114, 48, 80, 9, 39, 66, 97, 36, 104, 33, 55, 51, 119)
- Position Gates Buttons
- Sequencer Triggers (PLAY, STOP, STEP, RESET)
- Sequencer Parameters (RATE, ADD, SCALE, START POS, LENGTH)

Sequencer View

The **Sequencer** is the most powerful **Module** in **TC-11**. It outputs user-set values at a periodic rate, and has controls for all playback functions. It can also be used to send note frequency values, and has the ability to create a note filter so your patch stays in whatever key or mode you choose.

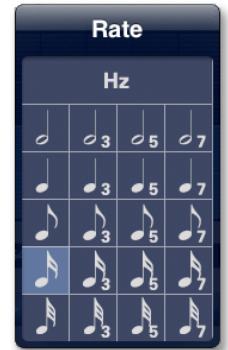


13.2 Minimize View Button

The **Minimize View Button** will hide the **Parameters** to shrink the view size. Press the button again to show the **Parameters**.

13.3 Rate Format Button

Choose the style that the **Rate Parameter** shows the **Sequencer** rate. You can choose Hz (cycles per second), or BPM (beats per minute) with the subdivision of your choosing. In BPM, the quarter note gets the beat. For example, choosing a sixteenth note, and setting the **Rate Parameter** to 60 BPM is equivalent to sending four (4) values per second.

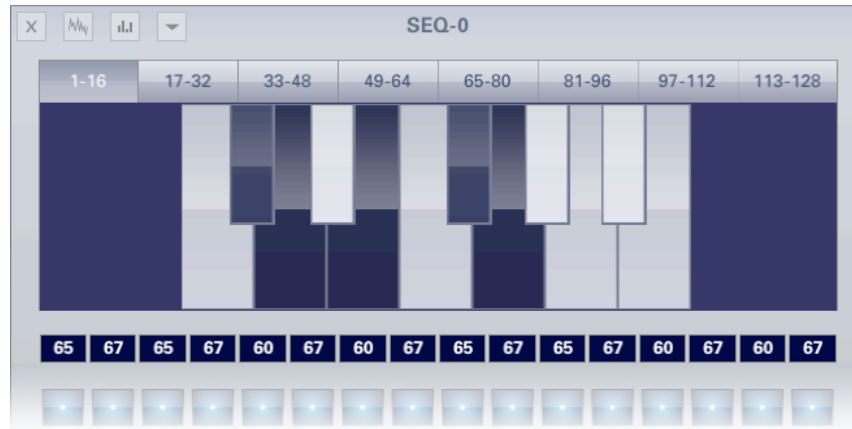
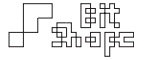


Note that choosing a different subdivision does not change the speed of playback, but rather updates the displayed BPM to reflect the new subdivision.

13.4 Keyboard Filter Button

When a **Parameter** is set to receive **Sequencer Values Are Notes** [Sequencer Values Are Notes Button, p. 38], the **Sequencer** values for that **Parameter** are treated as MIDI note numbers. This allows for tuned melodies. To keep the notes within a key when the **Sequence** is shifted or scaled, you can choose to filter out notes you do not want to hear. When a **Sequencer** value gets filtered, it will be rounded to the nearest acceptable note.

The **Keyboard Filter Button** shows the **Keyboard Filter View**. This is where you can choose which notes to allow and which to filter out.



Keyboard Filter View

Light colored notes are allowed through, and dark colored notes are filtered out. Tap any key to toggle its status. The filter settings are saved with your patch.

To return to the normal **Sequence Graph View**, press the **Keyboard Filter Button** again, or tap any of the **Sequence Banks**.

13.4 Randomize Values Button

The **Randomize Values Button** is a quick way to randomize the **Sequence**. When the button is pressed, you are given the option to randomize all **Sequence** values, or only the values within the current playback range.

13.5 Delete Button

The **Delete Button** will remove the **Sequencer** from the synthesis graph. Any **Parameters** that were controlled by the **Sequencer** will be reset to *Constant*. Any triggers that were controlled by the **Sequencer** will be set to *None*.



13.6 Sequence Banks

These buttons will show a sub-range of the full **Sequence**. The **Sequencer** can have a **Sequence** of up to 128 values. They are displayed in groups of 16, accessible through these buttons for editing.

13.7 Sequence Graph

This is the main editing area of the **Sequence**, and shows the values in a standard bar graph. Touch and drag any bar to change its value. You can any value, whether it is currently highlighted for playback or not. Keep in mind that the iPad's screen is multitouch enabled. You can edit multiple bars at the same time.

13.8 Sequence Values

The textfields below the **Sequence Graph** are the **Sequence Values**. Each textfield shows the value of that index of the **Sequence**. You can edit the value directly through these textfields.

Sequencer Values have a minimum value of 0, and a maximum value of 127. This range will be mapped to any **Parameter** that is controlled by the **Sequencer**, unless the **Parameter** is set to **Sequencer Values Are Notes** [Sequencer Values Are Notes Button, p. 38], in which case they will be treated as MIDI notes.

13.9 Position Gate Buttons

Position Gates are similar to mute buttons: they determine whether or not a **Sequence** index is allowed to be played. When the **Sequencer** steps to its next index, it checks whether or not the **Position Gate** is active. If it is active, the value is output, and a trigger is sent. If the **Position Gate Button** is unchecked, the **Sequencer** sends nothing. It will be heard as a pause, or rest, in the **Sequence**.



Use **Position Gates** to build a specific rhythm for the **Sequence**. For example, the triggers that are sent by the **Sequencer** can be used to start an **AHDSR**. Just by toggling **Position Gate Buttons**, you can create a rhythmic cycle for your **AHDSR**. You do not even have to set any **Sequence Values** in this case.

13.10 Sequencer Triggers

The **Sequencer** has four [4] trigger **Parameters**: *Play*, *Step*, *Stop*, and *Reset*. *Play* starts the **Sequence** playing, starting wherever it is currently positioned within the **Sequence**. Sending a second trigger to *Play* while it is playing does nothing. *Step* will instantly move the **Sequencer** to the next **Sequence** value. If *Step* is triggered while the **Sequencer** is currently playing, it will jump to the next value and continue moving forward at the playback rate.

Stop pauses **Sequencer** playback, and holds the current position. *Reset* will jump the playback position to the first value. When this happens, the value is not sent. Triggering *Play* after a *Reset* will start with the first value. If *Reset* is triggered while the **Sequencer** is currently playing, it will jump to the first value and continue moving forward at the playback rate, starting with the first value.

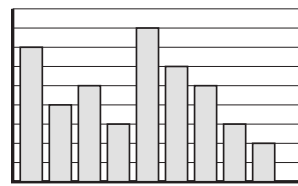
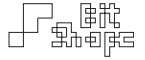
Part of the flexibility of the **Sequencer** is its playback control. Choosing which controllers you want to trigger playback movement is an important part of using the **Sequencer**.

Example:

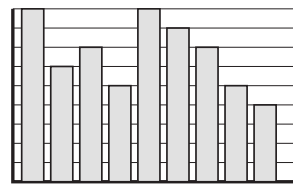
Load preset #50 - Martian Drums. The default setting plays the Sequencer when the patch is loaded. Change SEQ-0's PLAY trigger to None, and STOP to Patch Loaded. Then, the Sequencer steps forward only when you create a new touch.

13.10 Sequencer Parameters

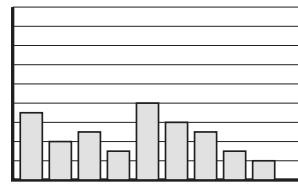
The **Sequencer** has five [5] **Parameters**: *Rate*, *Add*, *Scale*, *Start Position* and *Length*. *Rate* sets the playback rate in Hz. *Add* shifts all values by an equal amount. This value can be positive or negative to shift **Sequence Values** higher or lower. If the minimum or maximum value is reached, the value will be clipped. *Scale* changes values by a proportional amount.



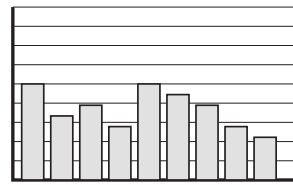
Original Values



Original Values, Add +2



Original Values, Scale 0.5

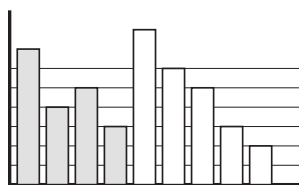
Original Values, Add +2,
then Scale 0.5

Sequencer Add and Scale Parameters

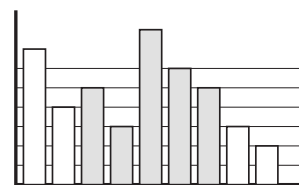
Add will be applied before *Scale*. A common use for *Add* is to shift up a melodic sequence, while *Scale* is more commonly used when a non-frequency based **Parameter** is being controlled by a **Sequencer**.

Start Position sets the first **Sequence** index for playback. When the **Sequencer** reaches its end, it will return to whatever the current *Start Position* is. *Length* is the length of the **Sequence**.

When playing back, a **Sequencer** will check to see if it is outside the currently defined **Sequence** range as set by the *Start Position* and *Length*. If it is, it will reset to the *Start Position* index.



Start Position is 1, Length is 4

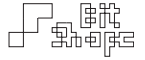


Start Position is 3, Length is 5

Start Position and Length Parameters

13.11 Sequencer Poly Routed

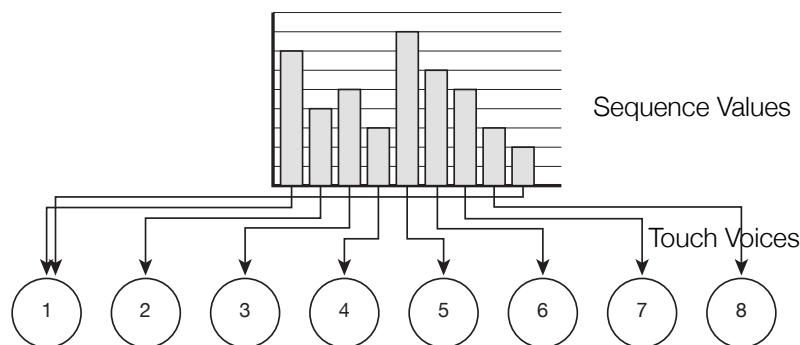
When controlling a **Parameter** with a **Sequencer**, there are three [3] choices for both normal and trigger controllers:



- SEQ-# Global
- SEQ-# Voiced
- SEQ-# Poly Routed

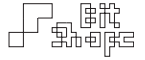
Global and voiced **Sequencers** act like any other controller: if a **Parameter** is controlled by the **Sequencer** in global mode, it will receive the same output values as all other **Parameters** that follow in global mode. Voiced mode essentially gives each touch voice its own **Sequencer** instance, so that each can have its own *Rate*, *Add*, and other **Parameters**.

Poly Routed mode is a special method of targeting touch voices with **Sequencer Values**. Each time the **Sequencer** steps to a new value, that value will be sent only to the next available touch voice. This way, all of the voices will be controlled by the same **Sequencer** instance (like global mode), but they only receive a single value until the polyphony has cycled around to that voice again.



In Poly Routed Mode, Sequence Values are sent individually to the next available touch voice

It is common to use this mode in conjunction with setting the **Sequencer Step** trigger to *Touch Began*. When a touch begins, the **Sequencer** will send off its new value that is specific for that voice. This allows for overlap of **Sequence Values**; you can hold one voice at a specific **Sequence** value while others are free to grab their own. This mode can yield some interesting performance results, since the length of the **Sequence** may not match the number of active touch voices, and voice **Parameters** that change in other parts of the **Patch** will affect individual **Sequence** values!



Recording

Recording Capabilities

TC-11 supports complete internal recording of its audio output. The captured audio is a 44.1kHz, 16 bit, stereo (.wav) file. You can record up to ten (10) minutes of audio. At the end of the ten minute session, a countdown timer will give you a ten (10) second warning before the recording automatically stops. Look for this timer beneath the **Recording Controls Button**.

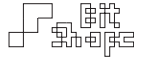
14.1 Recording Controls

To see the recording controls in the **Performance View**, set the **Show Recording Controls Button** option in the **Settings View**. When you want to see the recording controls, double-tap the **Recording Controls Button**.

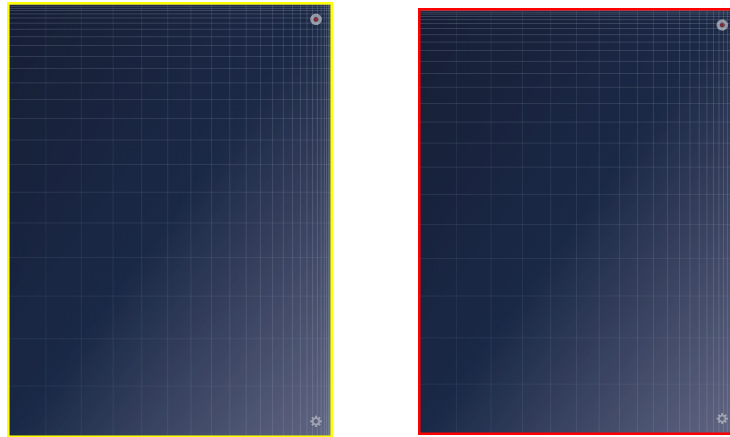


The recording controls. Double-tap the Recording Controls Button to access.

Record Start... does not immediately begin recording, but instead puts **TC-11** into a 'record ready' or 'record pause' mode. Recording begins exactly when the next touch begins in the **Performance View**. When in the 'record ready' mode, the **Performance View** will have a yellow



border indicating that the next touch will begin recording. Once that touch begins, the border will turn red indicating recording is active. Each recording overwrites the previous recording.



'Record ready' (left) has a yellow border. 'Recording active' (right) has a red border.

To stop the recording, double-tap the **Recording Controls Button** and press **Record Stop**. Recording can stop automatically a few ways:

- If ten (10) minutes elapse, a timer will appear giving ten (10) seconds notice, then recording will stop.
- If the view is changed away from the **Performance View**, recording stops.
- Exiting the app also stops recording.

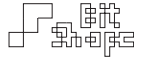
The **Play** option in the recording controls is available as long as a recording is presently available. It will play the currently saved recording so you can audition your performance. When playing, the button will become **Stop**, allowing you to stop the playback. You can play a saved recording even after changing to another patch.

The **AudioCopy** feature is an audio tool set by Sonoma Wire Works that allows seamless copying and pasting of audio files between iOS applications. While **TC-11** does not yet support AudioPaste, you can take your recording to any other **AudioCopy** supported app by pressing this button and following the prompt.



AudioCopy and AudioPaste are trademarks and software of Sonoma Wire Works.

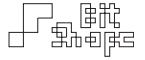
You can also access your recorded audio file by connecting your iPad to iTunes and looking in the app's Documents folder in the **File Sharing** section. The recorded audio file is named 'tc-11-temp-recording.wav.'



Appendices

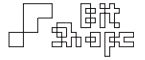
Troubleshooting

If you are experiencing odd behavior from **TC-11**, check the following common solutions for assistance. If your issue is not addressed here, please email [bitshapesoftware@gmail.com] with a description of the problem. If the issue is an app crash, please include the crash log, located at `~/Library/Logs/CrashReporter/MobileDevice/iPadName/TC-11_iPad.crash`. This will help keep **TC-11** as solid of an app as possible.



15.1 Issues / Solutions

<p><i>No sound</i></p>	<ul style="list-style-type: none"> • Check the iPad device volume. • Check receiver / amp / speaker volume if using the headphone out jack. • Try tilting the iPad during performance. Some patches require device motion to raise the patch volume. • Try using multiple touches to activate the patch. • Load / reload a different patch. • Quit / relaunch TC-11. Double-tap the home button, then tap and hold the TC-11 icon until the minus sign appears, then tap the minus sign. • If using a hardware DAC, try using the built-in output. If the issue is related to the DAC, please email the make and model to the developer. • Restart the iPad.
<p><i>Buzz or electrical noise during performance</i></p>	<ul style="list-style-type: none"> • Disconnect the USB cable from the iPad.
<p><i>Graphical slowdown</i></p>	<ul style="list-style-type: none"> • TC-11 has been optimized for audio synthesis, which takes priority over graphics. iPad 1 hardware may experience graphical slowdown. • Turn off <i>Override Controller Display</i> in the Settings View.



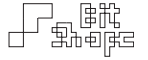
<i>Unwanted clicking in audio</i>	<ul style="list-style-type: none"> •The audio synthesis graph may be overloading the processor. •Simplify your patch by using fewer Modules. •Reduce the rate of playback for your Sequencers.
<i>Microphone is running through the speakers</i>	<ul style="list-style-type: none"> •Reset the iPad's General Settings, under General >> Reset >> Reset All Settings

Appendix A: Controllers

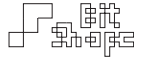
The term *First Touch* refers to the touch that begins a new group of touches. It will be colored differently than all following touches. The term *Previous Touch* refers to the last touch created before starting a new touch. The x-axis of the iPad is the narrow axis (768 pixels wide). The y-axis of the iPad is the long axis (1024 pixels).

16.1 Single Touch Controllers

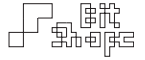
<i>Creation Time Since First Touch</i>	The time difference between the new touch beginning and the first touch beginning. Sent once when touch begins. Maximum: 5000 ms.
<i>Creation Time Since Previous Touch</i>	The time difference between the new touch beginning and the previous touch beginning. Sent once when touch begins. Maximum: 5000 ms.
<i>Touch Time Alive</i>	A running value that reports the time a touch has been active. Sent continuously. Maximum: 20000 ms.



<i>Touch Time Stationary</i>	A running value that reports the time a touch has not moved. Sent continuously. Maximum: 20000 ms.
<i>Touch Time Moving</i>	A running value that reports the time a touch has been moving. Sent continuously. Maximum: 20000 ms.
<i>Touch Time Alive When Ended</i>	A value that reports the time a touch has been active. Sent once when touch ends. Maximum: 20000 ms.
<i>Touch X Position</i>	The x-axis position of the touch. Sent continuously.
<i>Touch Y Position</i>	The y-axis position of the touch. Sent continuously.
<i>Touch X Position When Created</i>	The x-axis position of the touch. Sent once when touch begins.
<i>Touch Y Position When Created</i>	The y-axis position of the touch. Sent once when touch begins.
<i>Touch X Position When Ended</i>	The x-axis position of the touch. Sent once when touch ends.
<i>Touch Y Position When Ended</i>	The y-axis position of the touch. Sent once when touch ends.
<i>Touch Speed</i>	The speed of the touch. Sent continuously.
<i>Distance To First Touch</i>	The distance between the new touch position and the first touch position. Sent continuously.
<i>Distance To Previous Touch</i>	The distance between the new touch position and the previous touch position. Sent continuously.



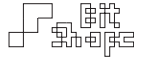
<i>Distance To Center</i>	The distance between the new touch position and the center of the screen. Sent continuously.
<i>Distance To Starting Position</i>	The distance between a touch and its original position on screen. Sent continuously.
<i>Distance Travelled</i>	A running value of the distance a touch has covered since it was created. Sent continuously.
<i>Angle To First Touch</i>	The angle between the new touch position and the first touch. The reference (0°) is fixed. Sent continuously.
<i>Angle To Previous Touch</i>	The angle between the new touch position and the previous touch. The reference (0°) is fixed. Sent continuously.
<i>Angle To Center</i>	The angle between the new touch position and the center of the screen. The reference (0°) is fixed. Sent continuously.
<i>Rotation Around First Touch</i>	The change of angle of the new touch around the first touch. The reference (0°) is always the original new touch position. Sent continuously.
<i>Rotation Around Center</i>	The change of angle of the new touch around the center of the screen. The reference (0°) is always the original new touch position. Sent continuously.
<i>Rotation Speed Around First Touch</i>	The rate of change of angle of the new touch around the first touch. The reference (0°) is always the original new touch position. Sent continuously.



<i>Rotation Speed Around Center</i>	The rate of change of angle of the new touch around the center of the screen. The reference (0°) is always the original new touch position. Sent continuously.
-------------------------------------	--

16.2 Group Touch Controllers

<i>Creation Time Since Previous Group</i>	The time difference between the first touch in a group beginning and the first touch of the last group beginning. Sent once when first touch begins. Maximum: 20000 ms.
<i>Group Time Alive</i>	A running value that reports the time a group has been active. Sent continuously. Maximum: 20000 ms.
<i>Group Time Stationary</i>	A running value that reports the time a group has not moved. Sent continuously. Maximum: 20000 ms.
<i>Group Time Moving</i>	A running value that reports the time a group has been moving. Sent continuously. Maximum: 20000 ms.
<i>Group Time Alive When Ended</i>	A value that reports the time a group has been active. Sent once when touch ends. Maximum: 20000 ms.
<i>Group Count</i>	The number of active touches. Sent when any touch begins.
<i>Touches Created Per Second</i>	The number of touches created over the last second. Maximum: 100.



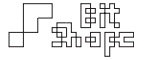
<i>Touches Ended Per Second</i>	The number of touches ended over the last second. Maximum: 100.
<i>Group Center X Position</i>	The average x-axis position of all active touches.
<i>Group Center Y Position</i>	The average y-axis position of all active touches.
<i>Group Total Speed</i>	The sum of all touches' speed.
<i>Group Center Speed</i>	The speed of the group's average position.
<i>Total Distance To First Touch</i>	The sum of all touches' distance to the first touch.
<i>Total Distance To Previous Touch</i>	The sum of all touches' distance to their previous touch.
<i>Total Distance To Center</i>	The sum of all touches' distance to the center of the screen.
<i>Total Angle To First Touch</i>	The sum of all touches' angle to the first touch.
<i>Total Angle To Previous Touch</i>	The sum of all touches' angle to their previous touch.
<i>Total Angle Around Center</i>	The sum of all touches' angle to the center of the screen.
<i>Average Angle To First Touch</i>	The average of all touches' angle to the first touch.
<i>Average Angle To Previous Touch</i>	The average of all touches' angle to their previous touch.
<i>Average Angle To Center</i>	The average of all touches' angle to the center of the screen.



<i>Total Rotation Around Center</i>	The sum of all touches' rotation around the center of the screen. Sent continuously, resets when group ends.
<i>Total Rotation Around Center Running</i>	The sum of all touches' rotation around the center of the screen. Sent continuously, does not reset when group ends.
<i>Total Rotation Speed Around Center</i>	The sum of all touches' rotation speed around the center of the screen.

16.3 Device Motion Controllers

<i>Accelerometer X</i>	The acceleration of the iPad along its x-axis. Minimum value at its resting point, maximum value at either extreme (positive or negative). iPad 2 and higher: filters out gravity.
<i>Accelerometer Y</i>	The acceleration of the iPad along its y-axis. Minimum value at its resting point, maximum value at either extreme (positive or negative). iPad 2 and higher: filters out gravity.
<i>Accelerometer Z</i>	The acceleration of the iPad along its z-axis. Minimum value at its resting point, maximum value at either extreme (positive or negative). iPad 2 and higher: filters out gravity.
<i>Gyroscope Pitch</i>	The device's rotation position around its x-axis. Minimum value at its resting point, maximum value at either extreme (positive or negative). Available iPad 2 and higher.



<i>Gyroscope Roll</i>	The device's rotation position around its y-axis. Minimum value at its resting point, maximum value at either extreme (positive or negative). Available iPad 2 and higher.
<i>Gyroscope Yaw</i>	The device's rotation position around its z-axis. Minimum value at its resting point, maximum value at either extreme (positive or negative). Available iPad 2 and higher.
<i>Compass Heading</i>	The magnetic heading of the device, pointing from the top of the device. The data stream is low-passed to smooth out jarring jumps. Available iPad 2 and higher.

16.4 Miscellaneous Controllers

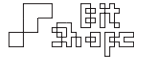
Constant	A fixed value.
----------	----------------

Appendix B: Triggers

The term 'trigger' is used to describe a command to start a synth process. The **AHDSR** and **Sequencer** modules use triggers to control playback. **Sequencers** can also generate triggers.

17.1 Single Touch Triggers

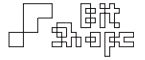
<i>Touch Began</i>	Trigger sent when a touch begins.
<i>Touch Ended</i>	Trigger sent when a touch ends.



<i>Touch Began Or Ended</i>	Trigger sent when a touch begins and when it ends.
<i>Touch Started Moving</i>	Trigger sent when a touch starts moving after being stationary.
<i>Touch Stopped Moving</i>	Trigger sent when a touch stops moving after being in motion.
<i>Touch Started Or Stopped Moving</i>	Trigger sent when a touch starts moving after being stationary, and when it stops moving after being in motion.

17.2 Group Touch Triggers

<i>Group Began</i>	Trigger sent when the first touch of a group begins.
<i>Group Ended</i>	Trigger sent when the last touch of a group ends.
<i>Group Started Moving</i>	Trigger sent when a group starts moving after all touches were stationary.
<i>Group Stopped Moving</i>	Trigger sent when a group stops moving after all touches were in motion.
<i>Touch # Began</i>	Trigger sent when a particular # touch begins in a group. For example, <i>Touch 3 Begins</i> sends a trigger when the third touch in a group begins.



<i>Touch # Ended</i>	Trigger sent when a particular # touch ends in a group. It will also send a trigger for touches of lower values. For example, <i>Touch 3 Ends</i> sends a trigger when the third touch in a group begins, and also the second and first.
----------------------	--

17.3 Sequencer Triggers

<i>SEQ-# Position Gates</i>	Trigger sent when a Sequencer steps to its next value. Global trigger.
<i>SEQ-# Position Gates Voiced</i>	Trigger sent when a Sequencer steps to its next value. Sent only to voiced controllers.
<i>SEQ-# Position Gates Poly Routed</i>	Trigger sent when a Sequencer steps to its next value. Sent only to the next available touch voice.
<i>SEQ-# Reset Trigger</i>	Trigger sent when a Sequencer steps to its first value. Global trigger.
<i>SEQ-# Reset Trigger Voiced</i>	Trigger sent when a Sequencer steps to its first value. Sent only to voiced controllers.
<i>SEQ-# Reset Trigger Poly Routed</i>	Trigger sent when a Sequencer steps to its first value. Sent only to the next available touch voice.

17.4 Miscellaneous Triggers

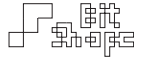
<i>None</i>	No trigger.
-------------	-------------



<i>Patch Loaded</i>	Trigger sent once when the user switches to the Performance View .
---------------------	---

Appendix C: Synth Objects

<i>OSC A</i>	Single wavetable oscillator. Can be paired with <i>OSC B</i> to have two simultaneous separate oscillators. Voiced.
<i>OSC B</i>	Single wavetable oscillator. Can be paired with <i>OSC A</i> to have two simultaneous separate oscillators. Voiced.
<i>FM OSC</i>	Frequency modulated wavetable oscillator. Voiced.
<i>Multi OSC</i>	Stacked wavetable oscillators. Up to five [5] oscillators, each transposed equally above the previous. Voiced.
<i>White Noise</i>	Random energy, equal energy per frequency band. Voiced.
<i>LPF Noise</i>	Random energy, equal energy per frequency band, filtered through a low-pass filter. Voiced.
<i>BPF Noise</i>	Random energy, equal energy per frequency band, filtered through a band-pass filter. Voiced.
<i>HPF Noise</i>	Random energy, equal energy per frequency band, filtered through a high-pass filter. Voiced.
<i>LPF</i>	Low-pass filter with resonance. Voiced.



<i>BPF</i>	Band-pass filter with resonance. Voiced.
<i>HPF</i>	High-pass filter with resonance. Voiced.
<i>Amp Mod</i>	Amplitude modulation using a selectable wavetable. Voiced.
<i>Amp</i>	Amplitude control. Voiced.
<i>Pan</i>	Panning control. Voiced.
<i>Downsampler</i>	Sample and hold filter with adjustable sample rate. Mix.
<i>Waveshaper</i>	Distortion through mapping incoming signal to a selectable wavetable. Mix.
<i>Phaser</i>	Rate variable allpass filter
<i>Mix LPF</i>	Low-pass filter with resonance. Mix.
<i>Mix BPF</i>	Band-pass filter with resonance. Mix.
<i>Mix HPF</i>	High-pass filter with resonance. Mix.
<i>Resonator</i>	Time-shifting variable delay line. Mix.
<i>Stereo Delay</i>	Variable delay lines for left and right channels. Mix.
<i>Reverb</i>	Recirculating delay reverb simulator