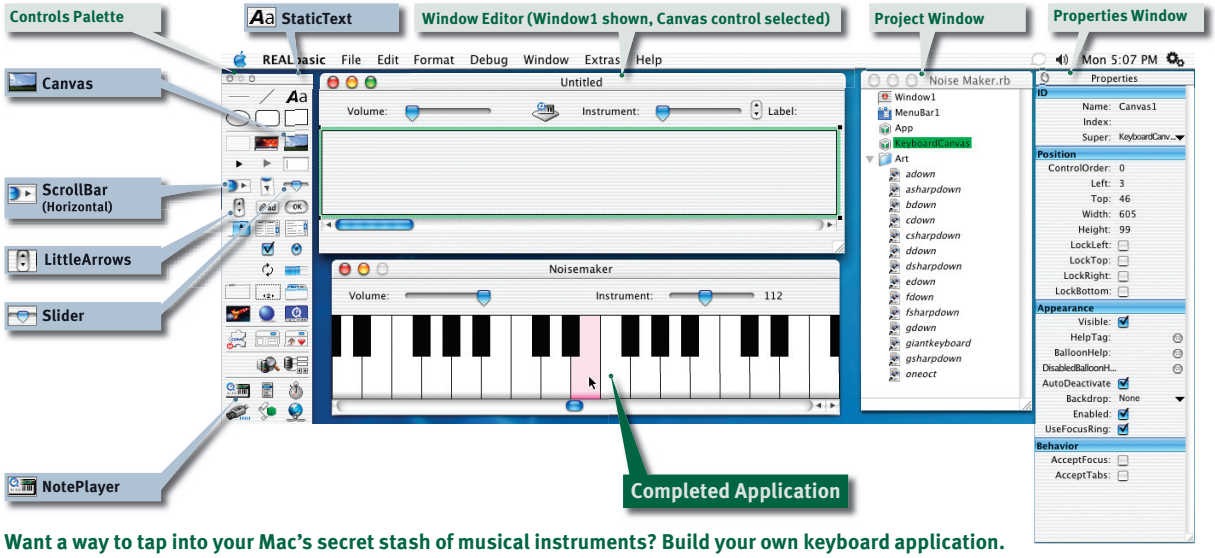
Build Your Own Music Maker

by Erick Tejkowski

TOUGH

WHAT YOU NEED

- Mac OS X ($129, www.apple.com)
- QuickTime 5 or later
- REALbasic 5 or later ($149.95, www.realsoftware.com)
- Music Project tutorial files (on the Disc)



Want a way to tap into your Mac's secret stash of musical instruments? Build your own keyboard application.

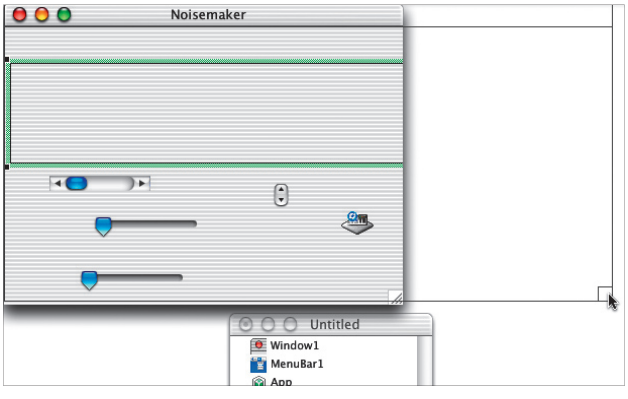
Pianos, guitars, sitars, tubas—musical instruments can be frivolous expenditures if you just want poke around but don't want to invest the time needed to learn them. Lucky for you, your Mac has a closet full of musical instruments stashed inside—you just need to coax them out to play. Wouldn't it be great to have an application that could put all of those instruments at your fingertips, playable from your own desktop keyboard? You can—and we show you how to build that app yourself.

ON THE DISC
REALbasic 5.2.1 demo and Music Project tutorial files

The key ingredient here is REALbasic. We show you how to use it to build a desktop piano-style keyboard that will let you control volume and instrument selection, and then use REALbasic's NotePlayer control to play a tune, using one of 128 different instruments and sound effects from Apple's QuickTime Music Synthesizer (see "What's That Sound?", p70, for a complete list of instruments).

1 Gain Some Control

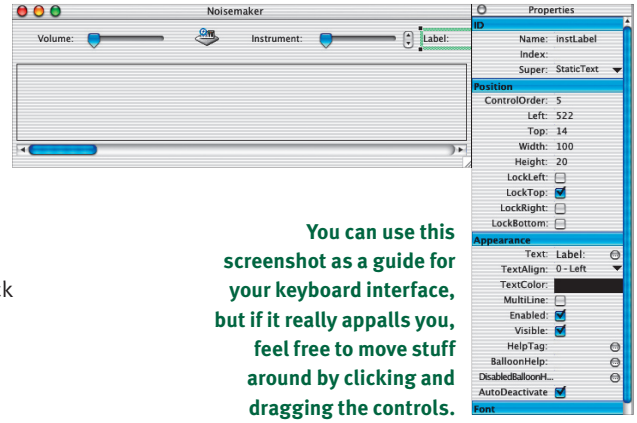
Launch REALbasic to start a new project. In the properties window for Window1, type a name for your application's interface in the Title field. From the controls palette, drag a NotePlayer, a Canvas, a horizontal ScrollBar, a LittleArrows, and two Slider controls into the window editor (the empty Untitled window that will become your keyboard interface). Click the Canvas1 control in the window editor. Change Width to 596 and Height to 99 in the properties window, and then resize the window to fit. ScrollBar1 will scroll the keyboard so you can play different keyboard octaves—click its control in the window editor, and set Width to 596 in the properties window. Slider1 will adjust volume and Slider2 will change the instrument—click each control and set Maximum to 127 in the properties window.



After changing the Canvas1 control's width and height, resize the window so it displays all controls fully.

2 Build the Interface

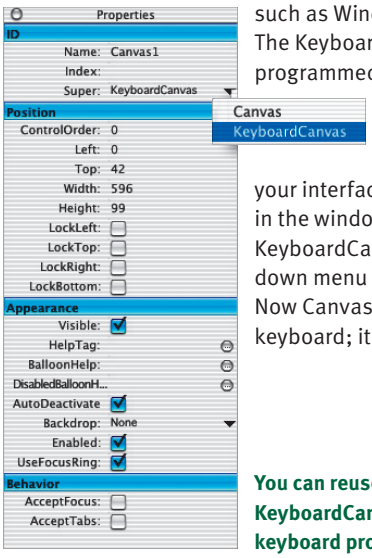
Drag three StaticText controls from the controls palette into the window editor. Click StaticText1, and in the properties window, type *instLabel* in the Name field and check the Visible box; this control will display the current instrument number used by the NotePlayer (the default is 1, a piano sound). Drag this to the right of the LittleArrows control, which will allow you to change instrument numbers incrementally. The remaining two StaticText controls will label the sliders. Click StaticText2 and in the properties window, type *Volume:* in the Text field and check the Visible box. Click StaticText3, type *Instrument:* in the Text field, and check the Visible box. Then position all controls in your project so they look like those in our screenshot (right), and save your project in its own folder as a REALbasic Standard Project.



You can use this screenshot as a guide for your keyboard interface, but if it really appalls you, feel free to move stuff around by clicking and dragging the controls.

3 Add Some Class

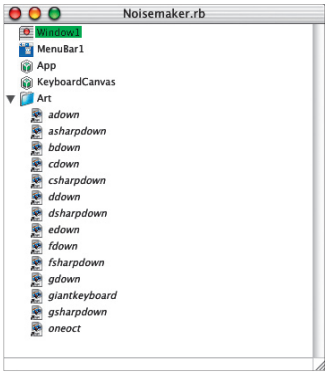
One of the great things about REALbasic is that you can use practically any *class* (a template that describes how an object behaves) from another project with your current one—we created a custom keyboard class for you, so you don't have sweat it out on your own unless you want to. Grab KeyboardCanvas from the Disc, and drag it into the project window (the one that holds project components such as Window1 and App) to add it. The KeyboardCanvas class is already programmed to trigger note values. By using it, you gain the benefits of the class without the hassle of recreating it. To tie it to your interface, click the Canvas1 control in the window editor, and then select KeyboardCanvas from the Super drop-down menu in the properties window. Now Canvas1 has the functionality of a keyboard; it just doesn't look like one yet.



You can reuse our homemade KeyboardCanvas class in your own keyboard project to play music.

4 Design Is Key(board)

Our KeyboardCanvas class calls upon a collection of 14 PICT images to create the look of a piano keyboard. One image displays all 128 keys on the keyboard. Another displays a single keyboard octave. The remaining twelve show a keyboard octave with a particular key pressed. We included all 14 images on the Disc to save you *a lot* of time. Feel free to customize the images in your favorite graphics application. To add these graphics to your project, copy the Art folder from the Disc (in the Music Project folder) to the same folder that holds your project. Then drag the Art folder from your project folder to the project window to add them.

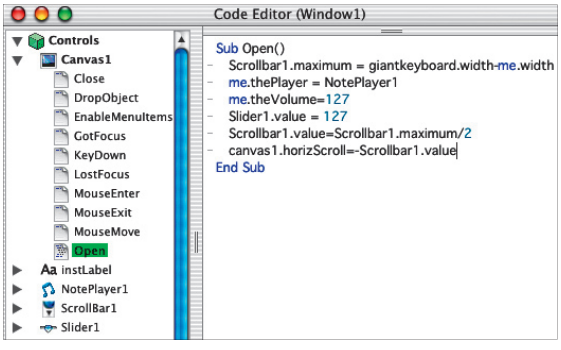


Once you drag our keyboard images into your project window, the KeyboardCanvas class can access them for display.

5 Code the Canvas

To make Canvas1 follow the settings dictated by the NotePlayer, ScrollBar, and Slider controls you created, you need to do a little coding. In the window editor, double-click the Canvas1 control to open the Code Editor window. Under the Canvas1 set of events, click the Open event and add the following code between the Sub Open and End Sub lines:

```
Scrollbar1.maximum = giantkeyboard.width-me.width
me.thePlayer = NotePlayer1
me.theVolume=127
Slider1.value = 127
Scrollbar1.value=Scrollbar1.maximum/2
canvas1.horizScroll=-Scrollbar1.value
```



This bit of code tells your keyboard app how to display the scrollbar, that it should use NotePlayer1 for sound, and where to set the volume slider when the app is launched.

6 Add Control to the Controls

Because the entire piano keyboard display is very large (128 keys total), the ScrollBar control will enable you to horizontally scroll through the lowest to the highest octaves on the keyboard display. Click the ScrollBar1 control in the window editor, and then check the LiveScroll box at the bottom of the properties window. Double-click the ScrollBar1 control to bring up the Code Editor, click the ValueChanged event to select it, and add the following code between the Sub lines:

```
Canvas1.horizScroll=-me.value
Canvas1.draw Canvas1.graphics
```

To program Slider1 to control volume, click the Slider1 triangle in the Code Editor window to display its events, click ValueChanged, and type `Canvas1.theVolume = me.value` between the Sub lines. To enable Slider2 to access the 128 instrument sounds, display Slider2's events, click ValueChanged, and enter this code between the Sub lines:

```
Canvas1.thePlayer.Instrument = me.value
instLabel.text = str(me.value)
```

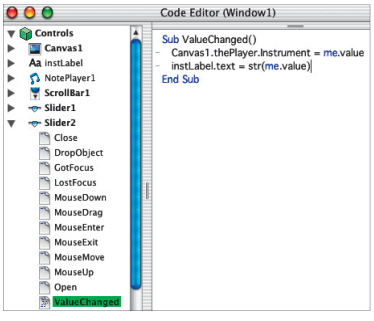
To enable the LittleArrows control to increase the instrument number by one, display its events in the Code Editor, click the Up event, and type the following code between the Sub lines.

```
if slider2.value<slider2.maximum then
slider2.value=slider2.value+1
else
slider2.value=slider2.minimum
end if
```

Then click its Down event and type the following code to enable it to decrease the instrument number by one.

```
if slider2.value>slider2.minimum then
slider2.value=slider2.value-1
else
slider2.value=slider2.maximum
end if
```

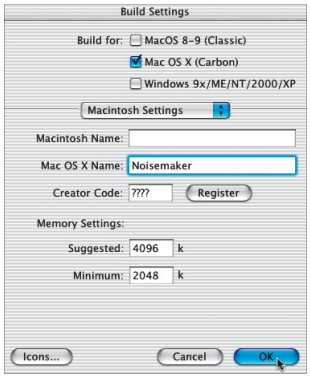
Now close the Code Editor window and save your project.



This bit of code enables you to select one of 128 built-in instruments using the second (Instrument) slider.

7 Debug, Tickle, and Build

The moment of truth arrives. From the Debug menu, select Run. REALbasic compiles your app and then launches it. Test it out by tickling the ivories (and ebonies), moving the Volume slider, and changing instruments. If everything looks, works, and sounds good, quit the test application, switch back to your REALbasic project, and select Build Settings from the File menu. In the resulting dialog, you can choose any platform on which to build your app (Classic Mac OS, Mac OS X, Windows, or any mix of these). Since our project was designed for OS X, check only the Mac OS X (Carbon) box. From the pop-up menu, select Macintosh Settings, and then give your masterpiece a name in the Mac OS X Name field. Click OK to build the settings. Then select Build Application from the File menu to bring your app to life and play.



REALbasic gives you the choice of building for Mac OS X, Classic Mac, and even Windows (if you dare go there). For this project, choose Mac OS X.

Erick Tejkowski is president of the Bring Back the Keyboard Power Button Club (BBKPBC). Unfortunately, the club currently has only one member.

WHAT'S THAT SOUND?

When you fire up your new keyboard app and start tinkering away, the default instrument you hear is QuickTime's Acoustic Grand Piano (instrument 1). But you've got another 127 diverse noisemakers at your fingertips. Here's the complete list of all instruments in QuickTime's synthesizer and their numbers for you to dial up in your spiffy new app.

Piano	Chromatic Percussion
1—Acoustic Grand Piano	9—Celesta
2—Bright Acoustic Piano	10—Glockenspiel
3—Electric Grand Piano	11—Music Box
4—Honky-Tonk Piano	12—Vibraphone
5—Electric Piano 1	13—Marimba
6—Electric Piano 2	14—Xylophone
7—Harpsichord	15—Tubular Bells
8—Clavinet	16—Dulcimer
Organ	**Guitar**
17—Drawbar Organ	25—Nylon String Guitar
18—Percussive Organ	26—Steel String Guitar
19—Rock Organ	27—Electric Jazz Guitar
20—Church Organ	28—Electric Clean Guitar
21—Reed Organ	29—Electric Guitar Muted
22—Accordian	30—Overdriven Guitar
23—Harmonica	31—Distortion Guitar
24—Tango Accordian	32—Guitar Harmonics
Bass	**Strings**
33—Acoustic Bass	41—Violin
34—Electric Bass (fingered)	42—Viola
35—Electric Bass (picked)	43—Cello
36—Fretless Bass	44—Contrabass
37—Slap Bass 1	45—Tremolo Strings
38—Slap Bass 2	46—Pizzicato Strings
39—Synth Bass 1	47—Orchestral Strings
40—Synth Bass 2	48—Timpani
Ensemble	**Brass**
49—String Ensemble 1	57—Trumpet
50—String Ensemble 2	58—Trombone
51—Synth Strings 1	59—Tuba
52—Synth Strings 2	60—Muted Trumpet
53—Choir Aahs	61—French Horn
54—Choir Oohs	62—Brass Section
55—Synth Vox	63—Synth Brass 1
56—Orchestra Hit	64—Synth Brass 2
Reed	**Pipe**
65—Soprano Sax	73—Piccolo
66—Alto Sax	74—Flute
67—Tenor Sax	75—Recorder
68—Baritone Sax	76—Pan Flute
69—Oboe	77—Bottle Blow
70—English Horn	78—Skakuhachi
71—Bassoon	79—Whistle
72—Clarinet	80—Ocarina
Synth Lead	**Synth Pad**
81—Square Wave	89—Fantasy
82—Sawtooth Wave	90—Warm
83—Calliope	91—Polysynth
84—Chiffer	92—Choir
85—Charang	93—Bowed
86—Solo Vox	94—Metallic
87—Fifths Saw Wave	95—Halo
88—Bass & Lead	96—Sweep
Synth Effects	**Ethnic**
97—Rain	105—Sitar
98—Soundtrack	106—Banjo
99—Crystal	107—Shamisen
100—Atmosphere	108—Koto
101—Brightness	109—Kalimba
102—Goblins	110—Bagpipe
103—Echoes	111—Fiddle
104—Space	112—Shanai
Percussive	**Sound Effects**
113—Tinkle Bell	121—Guitar Fret Noise
114—Agogo	122—Breath Noise
115—Steel Drums	123—Seashore
116—Woodblock	124—Bird Tweet
117—Taiko Drum	125—Telephone Ring
118—Melodic Tom	126—Helicopter
119—Synth Drum	127—Applause
120—Reverse Cymbal	128—Gunshot