Clay Allsopp's Top 5 Tips for Working With RubyMotion

1. Use CocoaPods and don't reinvent the wheel. CocoaPods allows you to easily import Objective-C libraries into your RubyMotion apps and use them just like any other iOS API, sort of like RubyGems. The iOS SDK has been around a while longer than RubyMotion, and there are some very mature projects (like Nimbus and AFNetworking) that you can and should leverage whenever possible.

2. Debug apps on your device. Very recently, RubyMotion added a debugging tool you can use while your app is running on your device. This allows you to set breakpoints, inspect variables, and get the backtrace of some tricky crashes. It's not a full REPL like you can use on the simulator, but it really can help track down some particularly tricky problems. The RubyMotion developer docs are a great starting point.

3. Write your own RubyGems! In the book we use third-party libraries like BubbleWrap, but they don't cover every facet of iOS development (yet!). If you find yourself using an iOS-C API and thinking, "Hmm, this isn't very Ruby-like", then absolutely consider writing a small wrapper and sharing it with the world. You need only to move a few files around and add a `Gemspec` file.

4. Use TestFlight to distribute beta versions of your app. TestFlight is a service that allows you to send in-development copies of your app to users without hooking up their device to your computer, as well as collect crash reports and usage statistics if you desire. The RubyMotion team has come out with a small gem which allows you to upload your app to TestFlight with one simple command, and it's a real time-saver.

5. Tell the community what you're up to! The RubyMotion user group is very active and is continually collaborating on new libraries, tutorials, or just news announcements. So if you get your RubyMotion app on the App Store or publish some code on Github, definitely add a post. I hang around there too, and I'd love to hear if this book helped!