

Extracted from:

Programming Google Glass, Second Edition

Build Great Glassware Apps with the Mirror API and GDK

This PDF file contains pages extracted from *Programming Google Glass, Second Edition*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2014 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

The
Pragmatic
Programmers

Programming Google Glass

Second Edition

Build Great
Glassware Apps
with the Mirror
API and GDK



Eric Redmond

Edited by Jacquelyn Carter

Programming Google Glass, Second Edition

Build Great Glassware Apps with the Mirror API and GDK

Eric Redmond

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at <http://pragprog.com>.

For international rights, please contact rights@pragprog.com.

Copyright © 2014 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.

ISBN-13: 978-1-941222-18-8

Encoded using the finest acid-free high-entropy binary digits.

Book version: B1.0—October 22, 2014

Preface

Google Glass is the new wearable computer that everyone is talking about. Not only does Glass offer a head-mounted optical display and touch interface, but it's also programmable in two ways: by programming Glassware (Glass applications) using the HTTP-based Mirror API, and by creating native applications using the Glass Development Kit (GDK). This book covers both.

You'll get a glimpse of what Glass is and what it is not, and how users can interface with Glass. In Part One of this book you'll learn how to develop a Glass application fast, by using the Mirror API to manipulate timeline cards and menus, track a Glass's geolocation, create rich interactions by responding to user inputs, and capture or serve user images and videos. In Part Two you'll learn how to shape user experience with the GDK by interacting with Glass hardware, from voice-to-text inputs, to QR code reading with the live camera, to building your own video game with fine-grained sensor inputs. You'll see how to properly design new Glassware or update existing applications to become Glassware. This is *the* book to read if you want a shortcut to this brave new world.

What's the Big Deal with Glass?

Imagine a world where computers did not exist, but the abilities that computers provided did. Rather than pulling a phone out of your pocket to talk with someone, you'd simply speak their name and you'd be connected. Or instead of taking out a laptop to play a video game, you would merely ask to play a game and it would appear. Or rather than sitting in front of a television screen to watch your favorite movies, a panel would hover conveniently in the air visible only to you, or sharable with friends who wish to watch as well. The idea of Google Glass is not to add more technology to your daily life, but rather sit idly in the background, available when you need it. It's about getting technology out of your way, while still providing its benefits.

The first future-facing movie that I can recall containing consumer HUD (heads-up display) goggles was *Back to the Future 2*. This HUD was worn in the *future* year 2015 (I know, right?), not by a military commander or an air-ship pilot, but by young Marty McFly, Jr., as he sat with his family around the kitchen table. This was a consumer device capable of, at least, displaying text and accepting phone calls.

Although Glass is sometimes considered to be an augmented-reality device, it's better thought of as an ever-present optical interface for a mobile device. It's a self-contained computer, yes, but it is also most useful when paired with an external paired smartphone for some actions, such as geolocation or continuous Internet access. Glass is sometimes referred to as having the power of a smartphone available, without the inconvenience of digging in your pocket.

Is This Book for You?

This book is designed to help experienced developers quickly start writing Google Glass applications with the Mirror API. Although this book covers using the interface with Google's Java Mirror Client code, the Mirror API itself is an HTTP interface with support for many languages. This means that many of the lessons about the Mirror API itself can apply to languages beyond the simple Java client.

The pertinent code is covered in the book, and the rest can be downloaded along with the book (or from GitHub.¹)

You needn't be a Java expert to follow Part One of this book, but it can help to know your way around the syntax and Eclipse editor. You may also get more out of Part One if you're familiar with Google App Engine, although you can use any Platform as a Service (PaaS) or host your own Glassware applications. Part Two requires a much more in depth knowledge of Java.

What's in This Book?

This book is intended to be read sequentially, from the first to last chapter. It covers most of the Mirror API and the GDK, but it's not designed to be a reference book. There are plenty of online documents² for more detailed reference material.

-
1. <https://github.com/coderoshi/glassbook>
 2. <https://developers.google.com/glass/develop/>

Part One

After an introduction to Glass and two styles of programming Glassware in [Chapter 1, *Wrapping Your Head Around Glass*, on page ?](#), we will dive into the Mirror API web service. In Part One we work on slowly building up a complete Glassware, along the way using most of the components of the Mirror API. Although I'd recommend you read both the Mirror API and GDK parts to get a full appreciation of your Glassware options, you are free to skip straight to Part Two if you have no interest in the Mirror API service.

[Chapter 2, *The Google App Engine PaaS*, on page ?](#)

Here we set up our Mirror API development and deployment environments. Since the Mirror API must run as a web service, we need a web provider to host the application. We chose Google App Engine, just to keep it in the Google family. The Glassware we're creating is called Lunch Roulette, which randomly chooses a lunch option for you.

[Chapter 3, *Authorizing Your Glassware*, on page ?](#)

Using the Mirror API also requires that end users of your Glassware authorize your application to interoperate with their Glass. Here we follow the OAuth 2.0 steps necessary to authorize our application. OAuth can be complex, so we cover the minimum amount necessary to keep you moving.

[Chapter 4, *Building the Timeline*, on page ?](#)

Finally the wait is over—we dig into the actual Mirror API, starting with the Glass Timeline. We add and remove timeline items, and add custom menu options to Lunch Roulette.

[Chapter 5, *Tracking Movement and User Responses*, on page ?](#)

Learn how to subscribe to notifications of Timeline changes, such as when a user deletes a card, or chooses a custom menu option. We'll also track user movement by Geolocation. We'll leverage notifications to expand Lunch Roulette to choose a restaurant currently close to the Glass user.

[Chapter 6, *Making Glass Social*, on page ?](#)

Here we finish up Lunch Roulette by adding some social aspects, such as calling the chosen restaurant to make reservations, or sharing images with your Contacts, and the Lunch Roulette application. We also look into attaching assets, such as an image, to a user's timeline.

[Chapter 7, Designing for Glass, on page ?](#)

One of the most important aspects of learning Glass is how it changes the world of application design. With its smaller screen, lack of a touchscreen or keyboard, and alternative input options, you have to approach UX design differently.

[Chapter 8, Turning a Web App to Glassware, on page ?](#)

If you have an existing web application, this chapter is a useful exercise. We take an existing blog application, and build Glassware from it. This way we leverage existing code, but also allow users the option of interfacing with our blog using a web browser or Glass.

Part Two

The chapters in Part Two are designed to build up from simple to more complex examples. Unlike Part One, however, the examples in each chapter are self contained. You should follow these chapters in order, rather than jump around, since previous sections are sometimes referred to later.

[Chapter 9, Introducing the GDK, on page ?](#)

This chapter kicks off Part Two by outlining what the GDK is, and how it differs from the Mirror API. It also goes through the steps of setting up a GDK development environment, and how to sideload your own apps onto Glass.

[Chapter 10, An Android Introduction on Glass, on page ?](#)

You have to crawl before you can walk, so we'll start by crawling through the basics of Android development. Details about Android can, and does, fill several books on its own. So instead of a full treatment, we'll do a flyover of the parts you'll need to jump into GDK programming. We'll cover Android development basics, project layouts, and Android programming model objects like Intents, Views and Components.

[Chapter 11, Live Cards, on page ?](#)

The first GDK UI elements we'll introduce are Live cards. Live cards are interactive cards that are rendered in realtime, as opposed to the static variety rendered by the Mirror API. We'll create and launch a live card application to gradually display statistics about the inner workings of Glass, including a simple menu to close it.

[Chapter 12, *Advanced Rendering and Navigation*, on page ?](#)

There are many ways to render live cards, but for applications that need video quality we must interact directly with the Glass rendering surface. We'll take advantage of this more complex method of rendering high frequency live cards to write a party application that animates balloons over an image. We'll create a more complex interface with multiple scrollable cards, and take photos with the Glass camera.

[Chapter 13, *Voice and Video Immersions*, on page ?](#)

Next we cover the other GDK UI element option called immersions. Immersions provide the full power of an Android application by running outside the constraints of the Timeline experience. We'll learn about immersions by creating a text-to-speech application with geolocation capabilities. We'll also create a QR code reading application that captures and renders video realtime from the Glass camera.

[Chapter 14, *Creating an Immersive Game*, on page ?](#)

We'll finish up our GDK tour by writing a side-scrolling video game for Glass. The game will use the built-in gravity sensor and gesture bar to control a player, while playing background music and sound effects.

[Chapter 15, *Preparing For the Real World*, on page ?](#)

Once we have the GDK under our belts, we'll look into the hardware of Google Glass. Then we'll cover some details about the Android operating system, and how to hack it, debug and optimize your applications. We'll optimize one of our previous examples to make it run nice and smoothly.

[Chapter 16, *Turning an Android App to Glassware*, on page ?](#)

Finally, we'll take an existing Android application and convert it into a Google Glass app. This follows a similar pattern to [Chapter 8, *Turning a Web App to Glassware*, on page ?](#), where an existing application was introduced, the necessary user stories were extracted, and we redesigned the app to work for Glass. And like the Mirror API example, much of the app code will be reusable, allowing us to leverage existing investments.

Online Resources

You can download the code and other resources used in this book from the Pragmatic Bookshelf website or my GitHub repository.^{3 4} You are free to use this source code for anything you wish.

You'll also find the community forum and the errata-submission form on the Pragmatic site, where you can report problems with the text or make suggestions for future versions.

The official Google Mirror API is also an excellent resource for the most up-to-date changes in the API, as well as other viewpoints on creating Glassware.⁵

Getting Going

Wearable computers, like Google Glass, are a growing topic, getting larger by the day. We could have easily created a book twice this length on Glass concepts, the Mirror API, good design, musings on the future, and so on. It was a conscious decision to keep this book slim so you can get a quick head start on this future.

We're beginning an exciting new journey in software development and design. Let's take our first steps into it.

Eric Redmond

October 2014

-
3. pragprog.com/book/erpgg/programming-google-glass
 4. <https://github.com/coderoshi/glassmirror>
 5. <http://developers.google.com/glass/>