



DEVELOPING APPS FOR

GLASS 

Note: This eBook relies on and uses information from the Google Glass Developers site.

# Table of Contents

## GLASS

What is Google Glass? .....	3
Technology Basics .....	3

## Design for Google Glass

User Interface .....	4
Timeline .....	4
UI Elements .....	5
	6

## Types of Glassware on Google Glass

Mirror API Glassware .....	7
GDK Glassware .....	7
Hybrid Glassware .....	7
	7

## Mirror API .....

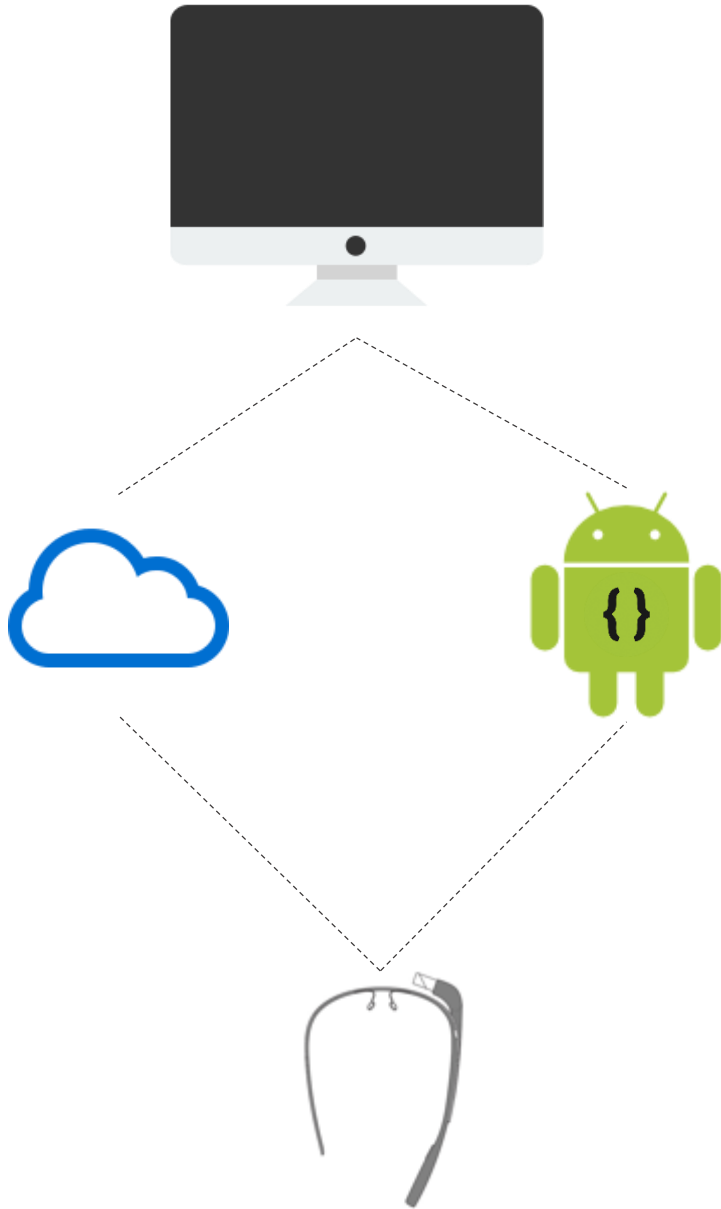
## Glass Development Kit (GDK) .....

## Develop, Install & Demo .....

## Tools .....

Playground .....	11
Glassware Flow Designer .....	11
Assets, Samples & Demos .....	11
	11

## Ask the experts .....



# GLASS

## What is Google GLASS?

Google Glass is a wearable computer with an optical head mounted display (OHMD). It is developed by Google Inc. Google Glass displays information similar to a smart phone, in a smaller screen size, using the optical head.

Wearers communicate with Glass via natural language voice commands or via gestures using the touchpad located on the side of the Google Glass.

## Technology Basics

Google Glass runs on the operating system Android 4.4.2 (KitKat).

Processor: Texas Instruments OMAP 4430 SoC 1.2Ghz Dual Core (ARMv7)

Memory: 2GB

Storage: 16GB (12 GB available)

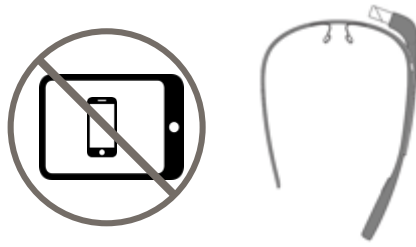
5-megapixel camera

WIFI and Bluetooth

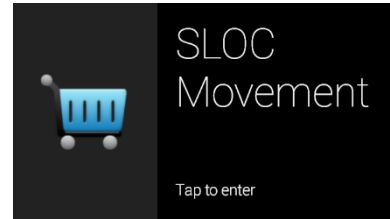




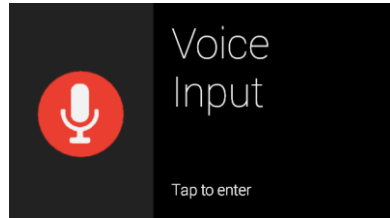
Design for Glass



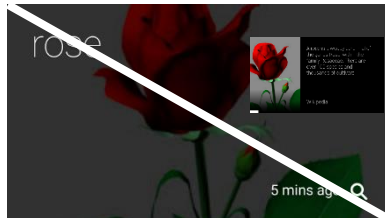
Don't get in the way



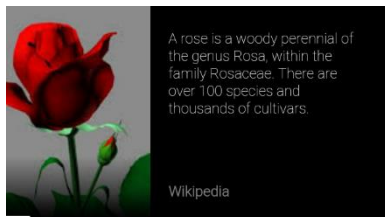
Keep it relevant



Avoid the unexpected



Build for people



## Design for Google Glass

### User Interface

Glass has a different user experience compared to a smart phone or a tablet. So Glass user interface has to be designed differently. It is necessary to understand the Glass user interface and interaction before understanding the development methods.

### Principles (As indicated by Google)

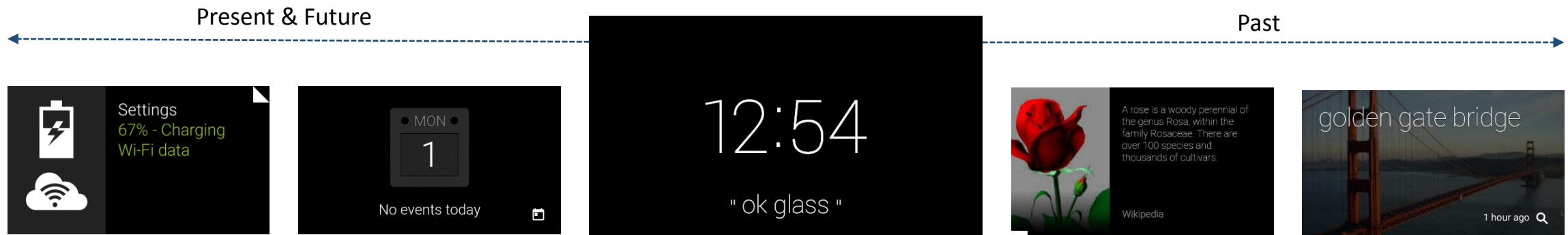
- **Design for Glass**
  - Glass is not same as smart phone or a tablet. Glass information has to be simple, relevant and current.
- **Don't get in the way**
  - Do not clutter the Glass and be consistent.
- **Keep it relevant**
  - Deliver contextually relevant data.
  - Relevant data increases engagement.
- **Avoid the unexpected**
  - Do not send unnecessary notifications.
  - Bad experience on Glass can turn away people.
- **Build for people**
  - Design with relevant images, natural language voice commands and natural gestures.
  - Make it easy for the users to use the Glass.

## Timeline

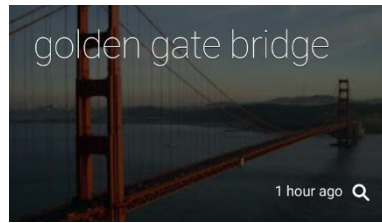
Timeline controls most of the user experience on Glass and is the main user interface that the users see on Glass. Timeline is a set of 640 \* 360 pixel cards. Timeline as it says is a line of cards. Timeline is organized into sections:

- **Home:** Home card is the default card in Glass and is the Glass clock. Home card resides at the center of the timeline. Users start on this card when the Glass wakes up. Users can also provide voice and touch commands to access Glassware.
- **Past:** Cards that appeared in the past are available to the right of the Home card (Glass clock). Past cards display static data and progressively pushed to the right indicating oldness. Cards are deleted when they are older than 7 days or when the limit of 200 cards are reached.
- **Present & Future:** Cards that display live information are available to the left of the Home card (Glass clock). Live cards display information that are relevant at the current time. Static cards that display information about the future (Ex: Calendar event) are also present in this section.

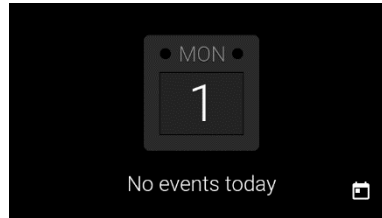
**Note:** 'Cards' and 'Timeline' are 'glassified' names for screens and a set of screens. If you have a difficult time relating to 'Cards' and 'Timeline' just consider each Card as a screen.



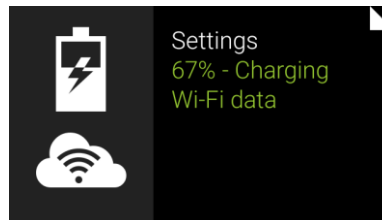
Live Card



Static Card



Settings



## Design for Google Glass

### UI Elements

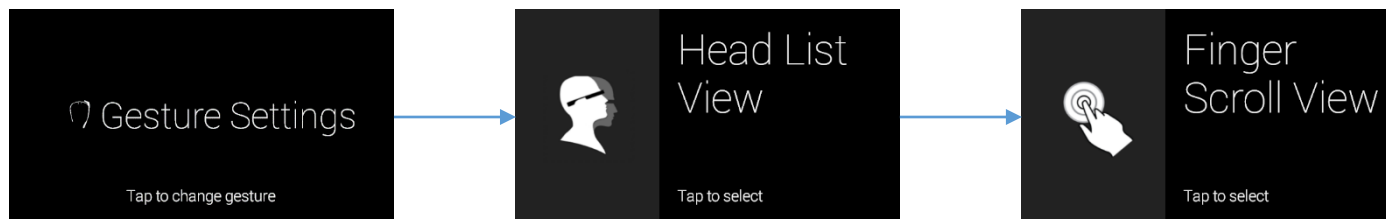
Timeline can contain the following:

- **Static Cards:** Static cards show information (text, images and video) that does not change at all or change infrequently.
- **Live Cards:** Show real-time information that are relevant at current time or at a future time.
- **Settings:** Settings is a set of cards that provide access to Glass settings. Ex: WIFI, Bluetooth, Enable Debug, etc.
- **Immersion:** Immersions run outside the timeline experience. Immersions are full processes that run on Glass as independent apps. Ex: Unvired Inventory Manager app on Google Glass
- **Menu Items:** Cards and immersions can contain menu items. Menu items carry out associated actions. Ex: Share, Delete, etc.

Immersion



Menu Items

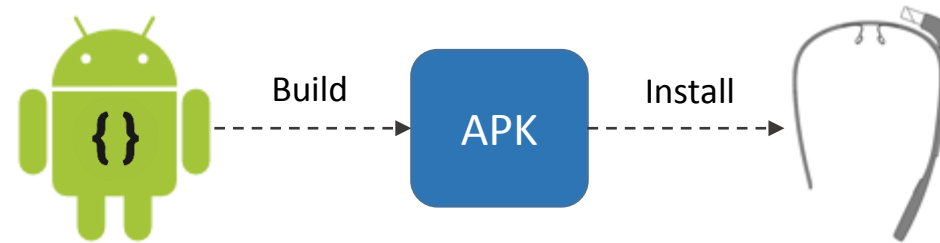


## Types of Glassware on Google Glass

Applications on Glass are called Glassware. Google Glass has 3 types of applications or interactions.

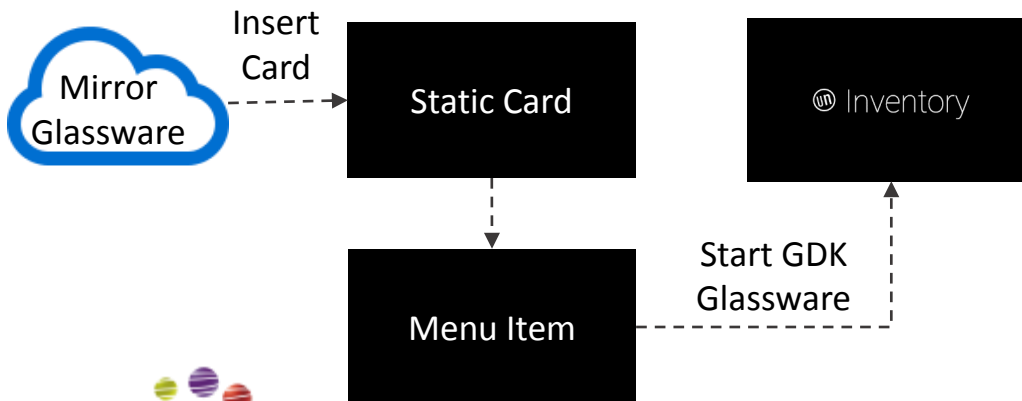
### Mirror API Glassware

Mirror API allows you to build web-based services that interact with Google Glass. Mirror API provides functionality over a cloud-based API and does not required running code on Glass. This is similar to the webpages in a browser.



### GDK Glassware

Glass Development Kit (GDK) is an add-on to Android SDK to build Glassware that runs directly on Glass as independent apps.



### Hybrid Glassware

Hybrid Glassware use both Mirror API and GDK. Mirror API Glassware can invoke GDK Glassware through menu items. This model can be used to push a card from Mirror API to the Glass and then opening the GDK Glassware from the card.

## Types of Glassware on Google Glass

### Mirror API Glassware

Mirror API allows you to build web-based services that interact with Google Glass. Mirror API provides functionality over a cloud-based API and does not require running code on Glass. This is similar to the webpages in a browser. Mirror API is a RESTful API (web based APIs) that provides the cards and menu items (actions) to the Glass. Mirror API can be used if you need:

- Platform Independence – Develop outside the Glass on a server in any supported language
- Common Infrastructure
- Built-in Functionality

### Mirror API Features

- **Static Cards:** Insert, modify and delete static cards. Objects like media and location can be attached to Static Cards.
- **Location:** Google Mirror API can observe the user's location in timeline items, request their last known location and provide contextual information to Glass.
- **Authorization:** OAuth 2.0 credentials have to be used to authorize requests to Google Mirror API.
- **Push:** Push static cards to Google Glass.

Mirror API Glassware can be developed in Java, PHP or Python.



Platform  
Independence

Common  
Infrastructure

Built-in  
Functionality



## Types of Glassware on Google Glass



Real-time User Interaction



Offline Functionality



Access to Hardware

### GDK Glassware

Glass Development Kit (GDK) is an add-on to Android SDK to build Glassware that runs directly on Glass as independent apps. Use GDK approach if you need:

- Real-time User Interaction
- Offline Functionality
- Access to Hardware

### GDK Allows

- **Live Cards**
- **Immersion**
- **Voice Input:** With voice user can use the Glass in a truly hands free manner.
- **Touch Gesture:** Glass supports tapping, swiping and scrolling gestures to provide input.
- **Location and Sensor:** Location and sensors (Accelerometer, Gravity, Gyroscope, Light, Linear Acceleration, Magnetic Field and Rotation Vector) are supported using the standard Android APIs.
- **Camera:** Camera can be used to capture images, video and read QR codes & barcodes.
- **Authentication:** GDK provides a web service to authenticate users against a web service. Users can enter their credentials when they install the Glassware. This is possible only if apps are listed in MyGlass.

## Develop, Install & Demo

### Mirror API

Mirror API Glassware can be developed in Java, PHP or Python.

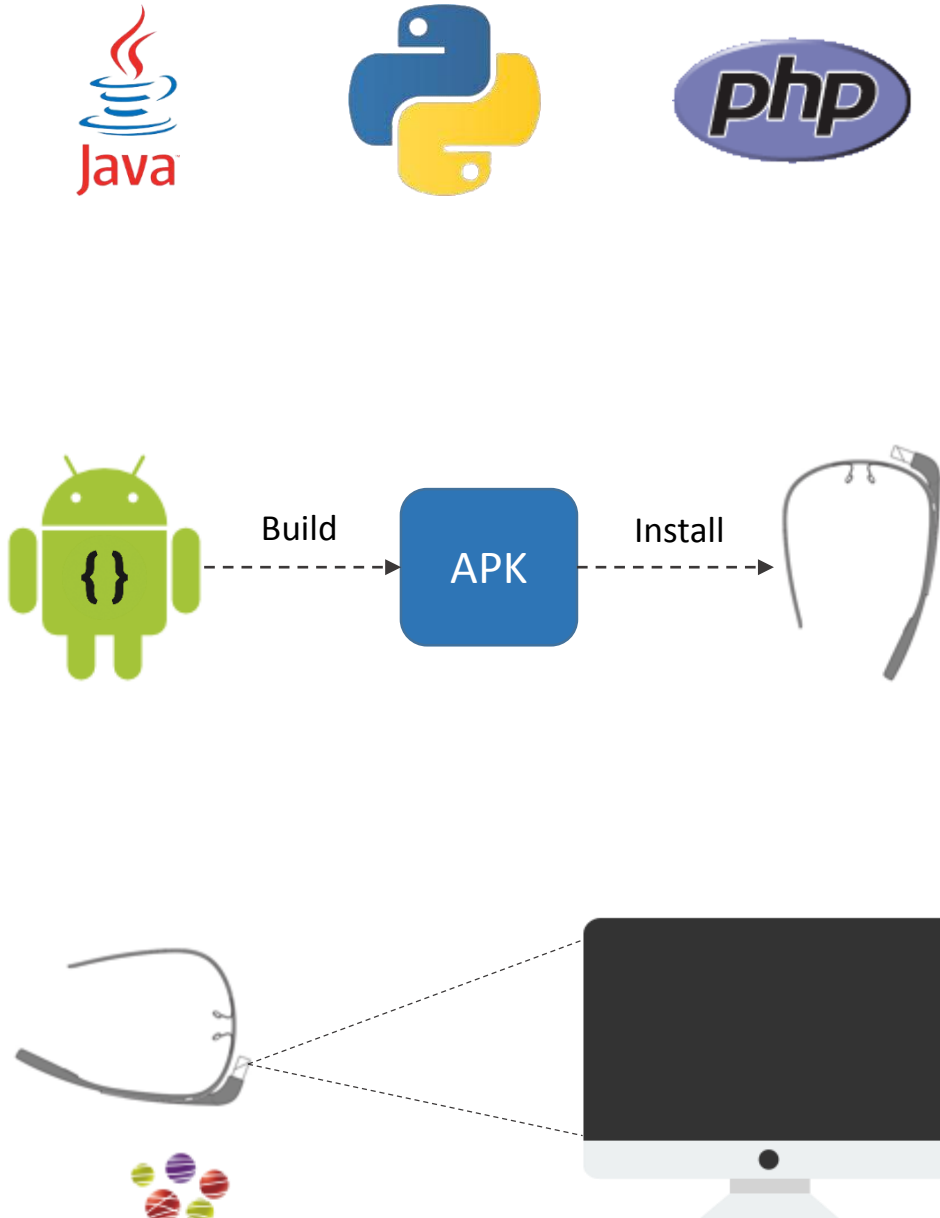
Mirror API Reference: <https://developers.google.com/glass/v1/reference/>

### GDK

1. GDK Glassware are developed in Java using the GDK add-on over Android SDK.
2. Android 4.4.2 (API 19) SDK and Glass Development Kit Preview add-on are required to develop GDK apps.
3. **GDK Reference:**  
<https://developers.google.com/glass/develop/gdk/reference/index>
4. Turn on Debug Mode in Google Glass from Settings to side-load and debug apps on Google Glass.
5. Connect Google Glass to the development system. In the ADT or Eclipse verify Glass is connected to the system in DDMS.
6. Select the project in ADT or Eclipse. Right click and use 'Run As' to run apps on Google Glass.

### Remote Display & Demo

Android does not support remote display of Google Glass on the PC or Mac to view the screen either in a larger size or for a demo. Use a free tool like 'Droid @ Screen' (<http://droid-at-screen.ribomation.com/>) to project Glass on the PC.



## Tools



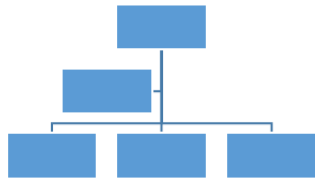
### Eclipse ADT

Eclipse Android Development Tools (ADT) IDE are required to develop GDK based apps. Android SDK and Eclipse ADT can be downloaded from: <http://developer.android.com/sdk/index.html>



### Playground

Google Mirror API Playground is a web based tool from Google that lets you experiment how the content is displayed on Google Glass. Access the Playground here: <https://developers.google.com/glass/tools-downloads/playground>



### Glassware Flow Designer

Glassware Flow Designer is a web based tool from Google to design the Cards and the navigation flow using common patterns and layouts. Glassware Flow Designer can be accessed here: <https://developers.google.com/glass/tools-downloads/glassware-flow-designer>

### Assets

Google provides assets like icons to help developers be consistent on the Glass and get a quick start. The Google Glass assets can be accessed here: <https://developers.google.com/glass/tools-downloads/downloads>

### Samples & Demos

Samples for Mirror API and GDK are available here: <https://developers.google.com/glass/samples/index>





**www.unvired.com**