CS 646 Android Mobile Application Development
Spring Semester, 2015
Doc 22 Background Tasks, Wear, Interface Design
Apr 29, 2015

Background Tasks Revisited

Running in the Background

Java Threads +
Activity.runOnUiThread(Runnable)
View.post(Runnable)
	Good for Simple tasks
	Get complex

AsyncTask	Good for one time task
Loopers + Handlers	Good for repeating tasks
Loaders	Good for reacting to events

Fetching a URL

```
public string fetchUrl(String url) {
    String contents = null
    try {
        URL url = new URL(url);
        URLConnection urlConnection = url.openConnection();
        int contentLength = urlConnection.getContentLength();
        InputStream in = new BufferedInputStream(urlConnection.getInputStream());
        byte[] buffer = new byte[contentLength];
        int bytesRead = 0;
        while (bytesRead < contentLength) {
            bytesRead +=  in.read(buffer, bytesRead, contentLength - bytesRead);
        }
        String contents = new String(buffer);
    } catch (Exception e) {
        Log.e("rew", "Bad", e);
    }
    return contents;
}
```

Using Thread

```
public void onClick(View v) {
    new Thread(new Runnable() {
        public void run() {
            final String instructor = fetchUrl("http://bismarck.sdsu.edu/rateme/instructor/2");
            textView.post(new Runnable() {
                public void run() {
                    textView.setText(instructor);
                }
            });
        }
    }).start();
}
```

Loaders

Added Android 3

Classes/Interfaces

Loader AsyncTaskLoader CusrorLoader	Subclass AsyncTaskLoader
LoaderManager	Manages lifecycle of Loader
LoaderManager.LoaderCallbacks	Interface How loader runs code on main thread

LoaderManager.LoaderCallbacks

Methods your class implements to receive data from Loader

onCreateLoader(int id, Bundle args)

Instantiate and return a new Loader for the given ID.

onLoadFinished(Loader<D> loader, D data)

Called when a previously created loader has finished its load.

onLoaderReset(Loader<D> loader)

Called when a previously created loader is being reset, and thus making its data unavailable.

Simple Example - Load Assignment 3 URL

Example does not use all lifecycle methods of Loaders

- Don’t have events

Main Activity

```
public class MainActivity extends ActionBarActivity implements
LoaderManager.LoaderCallbacks<String> {
    SampleLoader urlLoader;
    final static int loaderID = 0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Bundle arguments = urlBundle("http://bismarck.sdsu.edu/rateme/instructor/2");
        urlLoader = (SampleLoader)
                                getLoaderManager().initLoader(loaderID, arguments, this);
     }

    private Bundle urlBundle(String url) {
        Bundle arguments = new Bundle();
        arguments.putCharSequence("url", url);
        return arguments;
    }
```

Main Activity - LoaderCallbacks methods

```
//Called by LoaderManager to create new loader
public Loader<String> onCreateLoader (int id, Bundle args) {
    String url = (String) args.getCharSequence("url");
    return new SampleLoader(this, url);
}

public void      onLoadFinished(Loader<String> loader, String data) {
    Log.i("rew", "onLoadFinished " + data);
}

public void onLoaderReset(Loader<String> loader) {
    Log.i("rew", "onLoaderReset");
}
```

Main Activity - Button Clicked Method

```
public void restart(View source) {
    Log.i("rew", "restart");
    Bundle arguments = urlBundle("http://bismarck.sdsu.edu/rateme/list");
    getLoaderManager().restartLoader(0, arguments,this);
}
```

Loader Class

```
public class SampleLoader extends AsyncTaskLoader<String> {

    public String urlString;
    private String mData;

    public SampleLoader(Context activity, String url) {
        super(activity);
        urlString = url;
    }

    protected void onStartLoading() {
        forceLoad();
    }
```

Loader Class

```
public String loadInBackground() {
    String contents = "Not loaded";
    try {
        contents = fetchUrlContents(urlString);
    } catch (Exception e) {
        Log.e("rew", "Bad", e);
    }
    return contents;
}
```

Loader Class

```
private String fetchUrlContents(String urlToFetch) throws IOException {
    URL url = new URL(urlToFetch);
    URLConnection urlConnection = url.openConnection();
    int contentLength = urlConnection.getContentLength();
    InputStream in = new BufferedInputStream(urlConnection.getInputStream());
    byte[] buffer = new byte[contentLength];
    int bytesRead = 0;
    while (bytesRead < contentLength) {
        bytesRead +=  in.read(buffer, bytesRead, contentLength - bytesRead);
    }
    return new String(buffer);
}
```

For Full Details Read

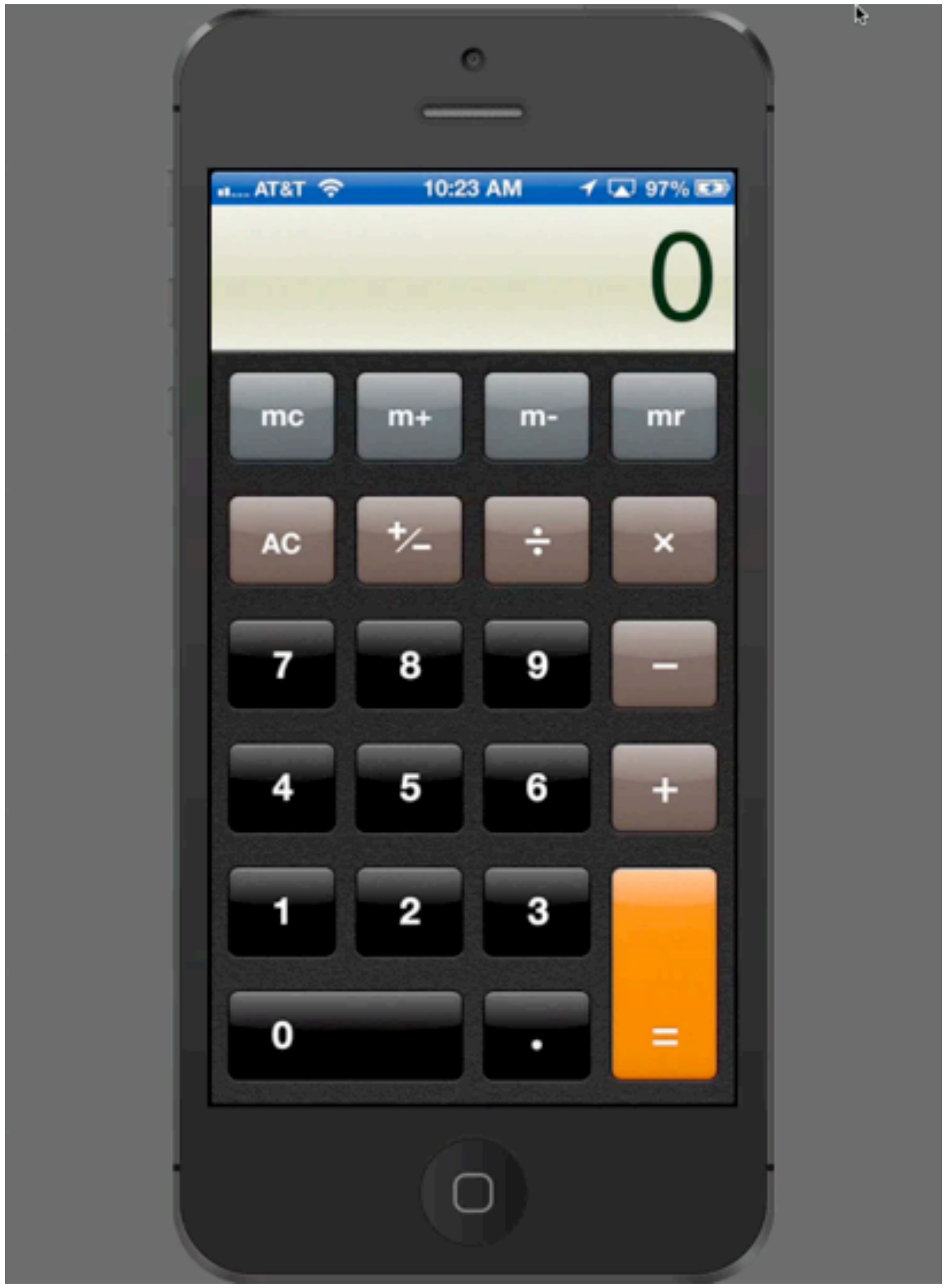
Big Nerd Ranch Chapter 35

Online tutorial

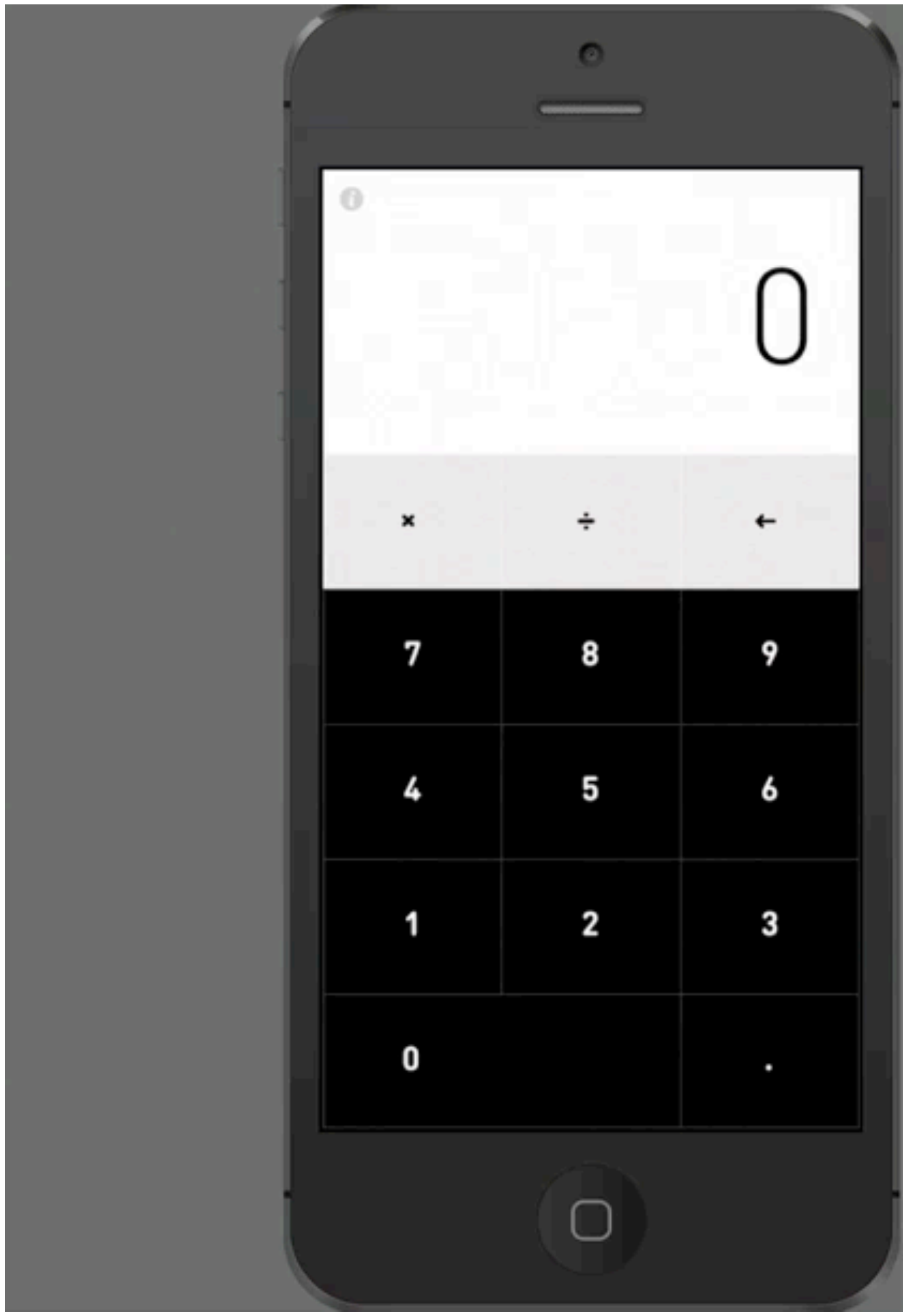
http://www.androiddesignpatterns.com/2012/08/implementing-loaders.html

User Interface

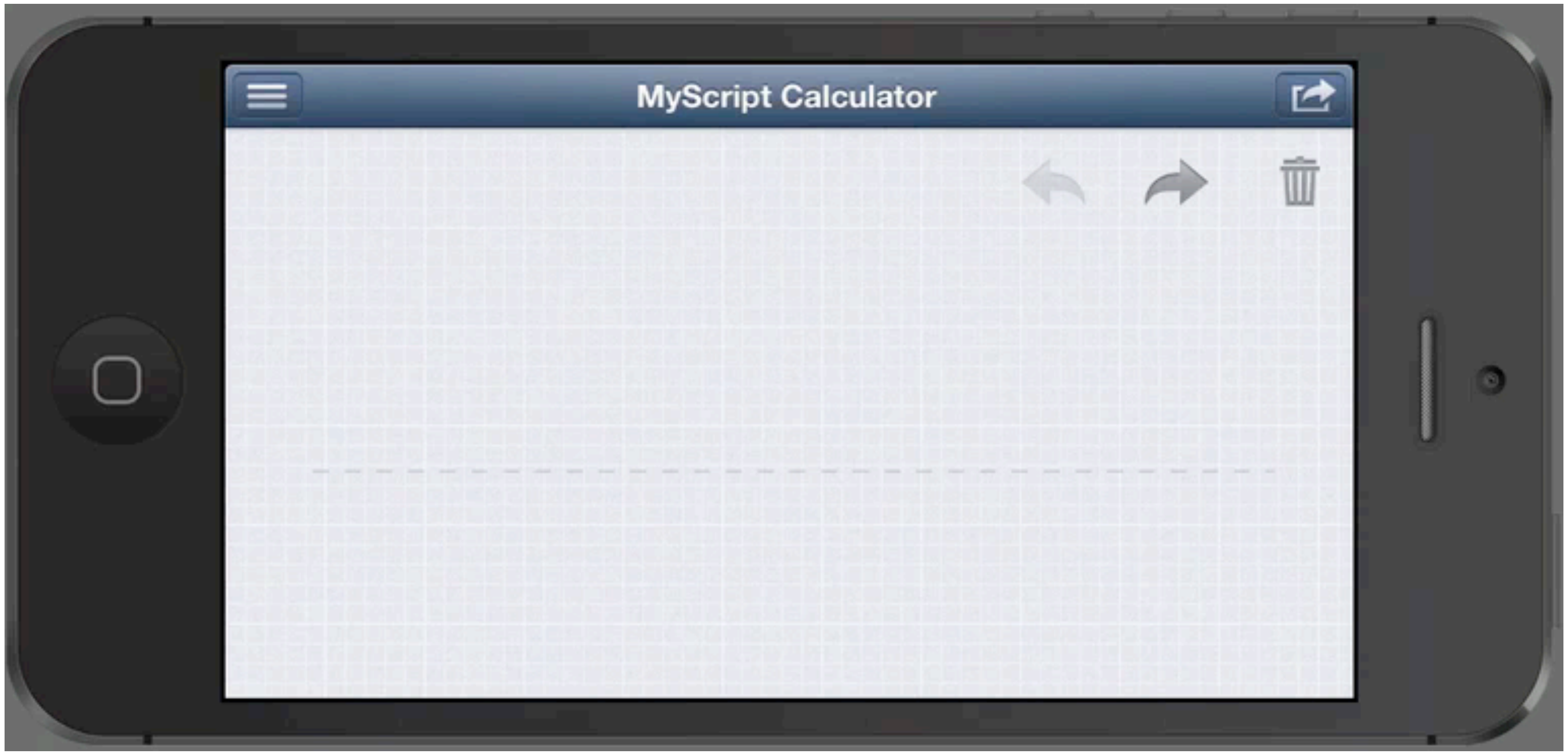
Calculator



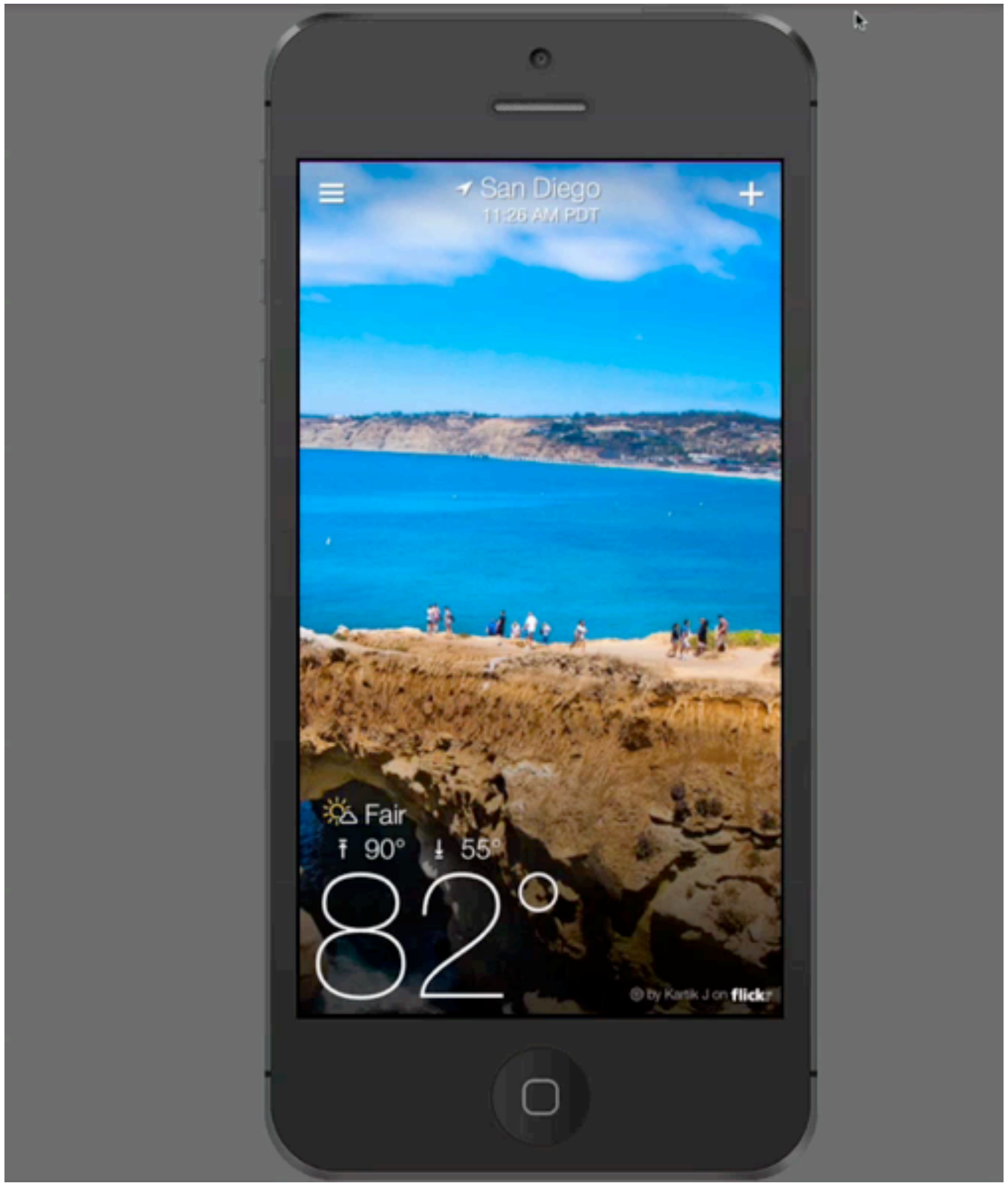
Rechner Calculator



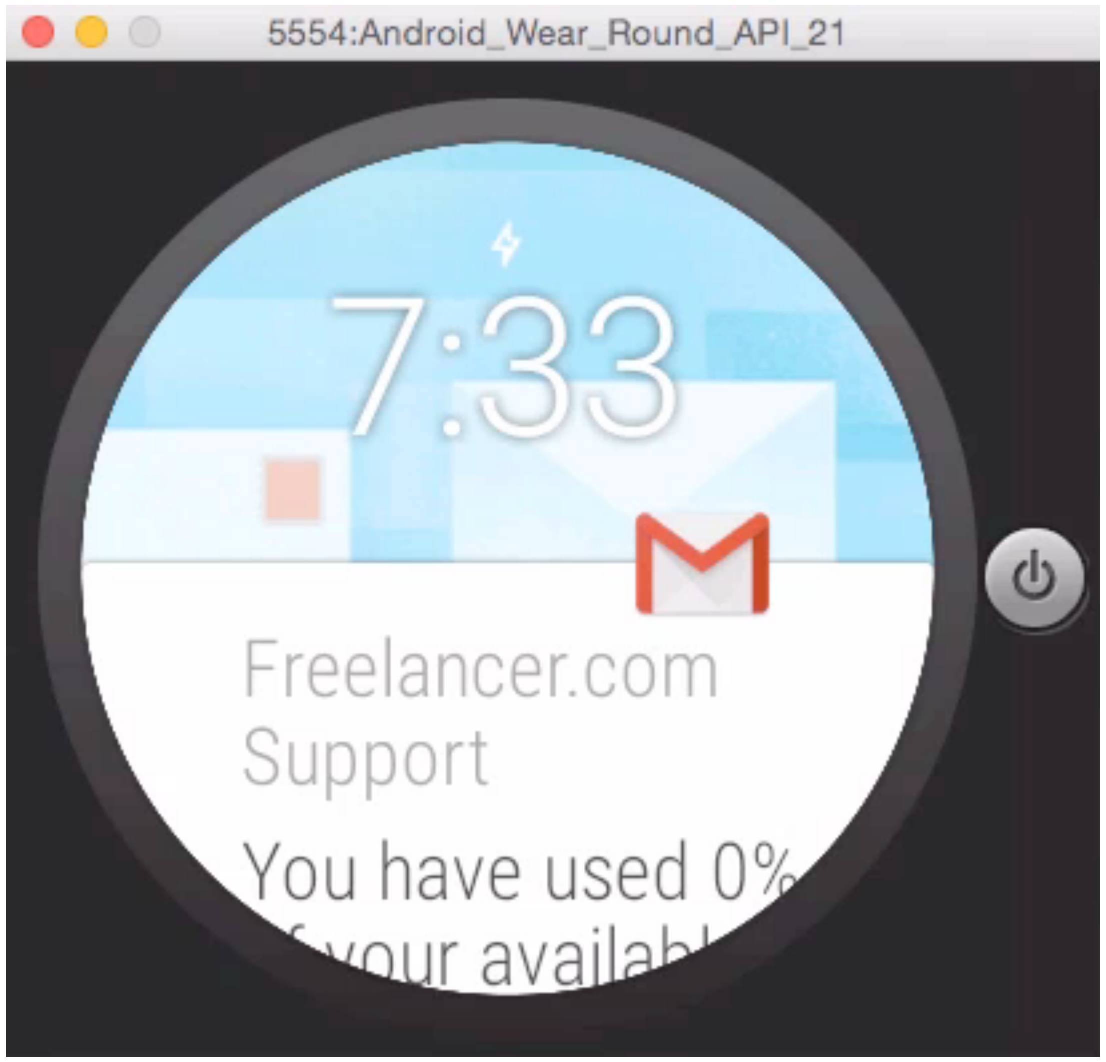
Script Calculator



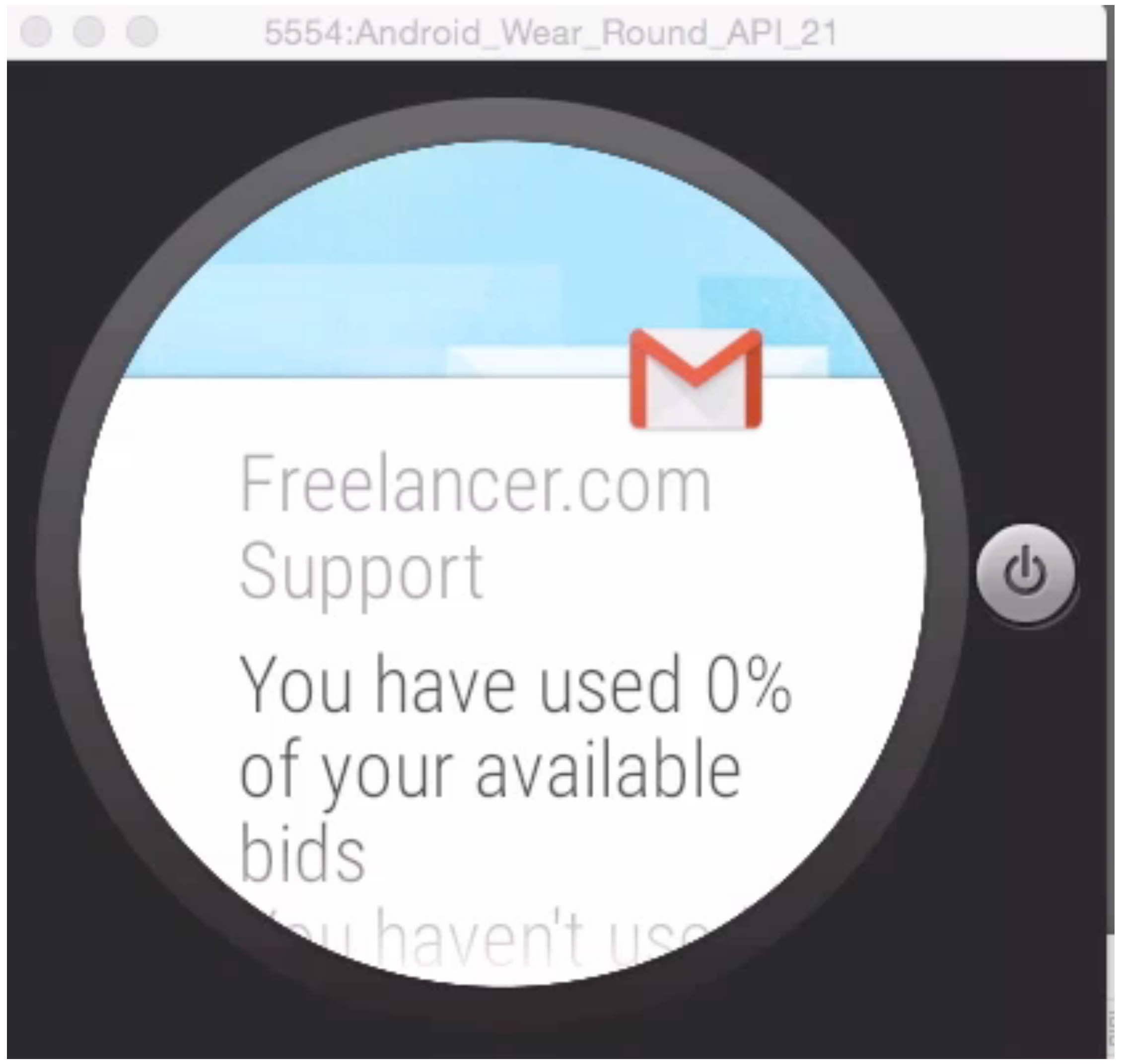
Yahoo Weather



Basic


5554:Android_Wear_Round_API_21
7:33
Freelancer.com
Support
You have used 0%
of your availab

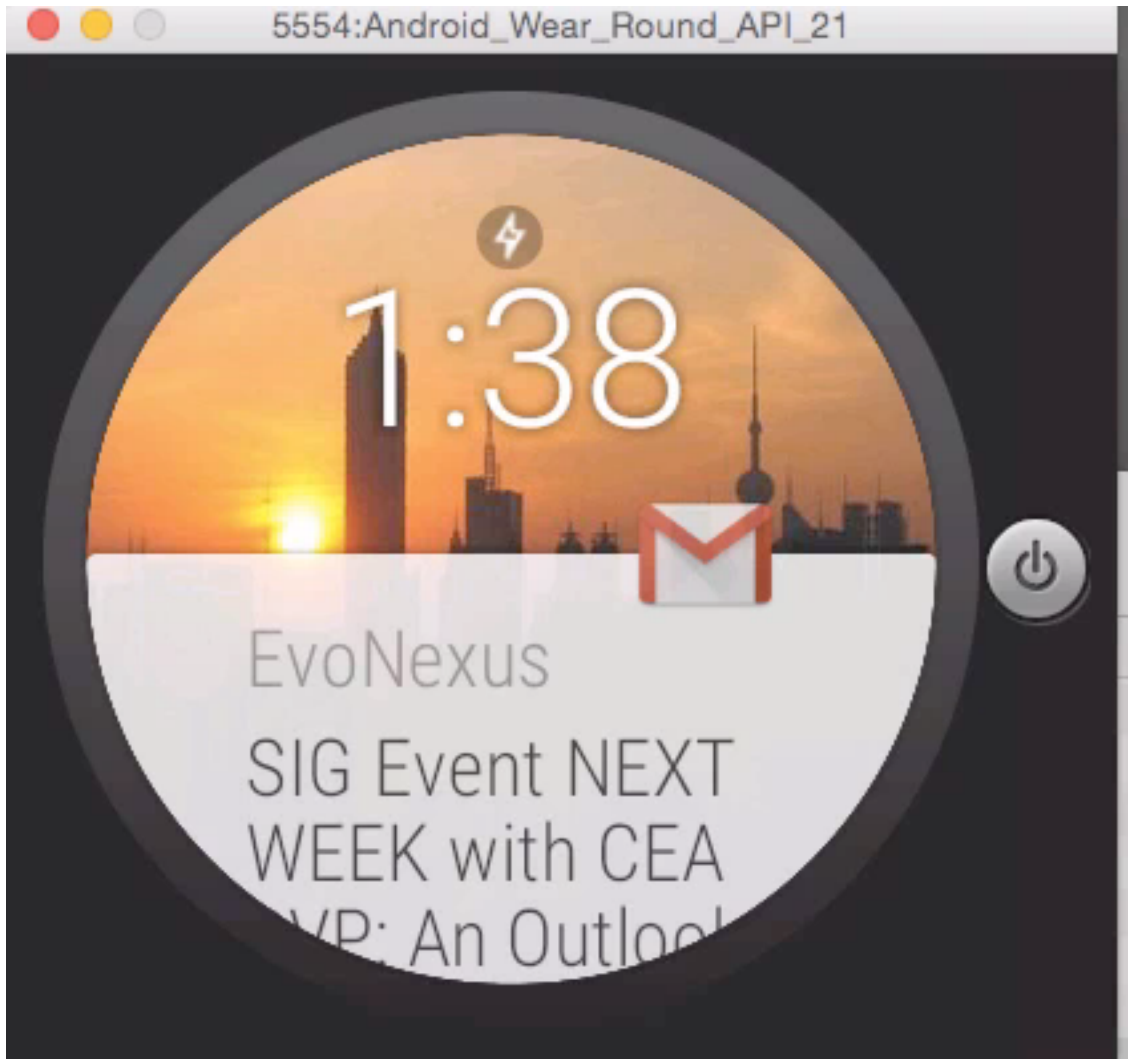
Phone Call


5554:Android_Wear_Round_API_21

Freelancer.com Support

You have used 0% of your available bids

Apps


5554:Android_Wear_Round_API_21
1:38
EvoNexus
SIG Event NEXT
WEEK with CEA
VP: An Outlo

Design

Focus on not stopping the user

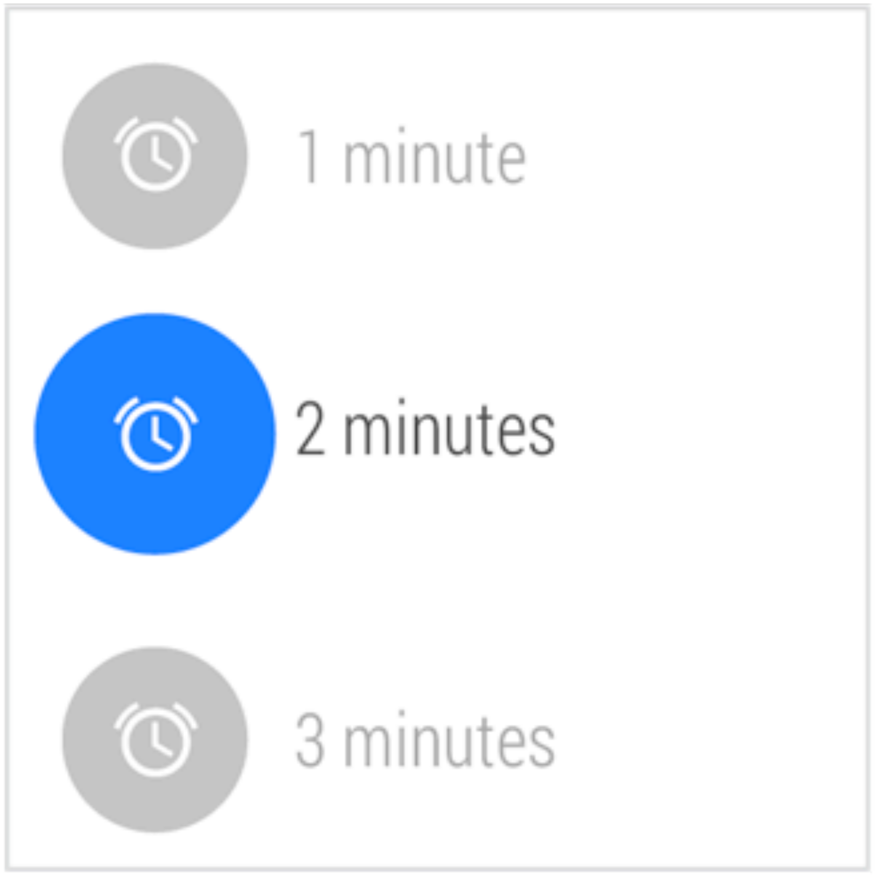
- It should take less than 5 seconds

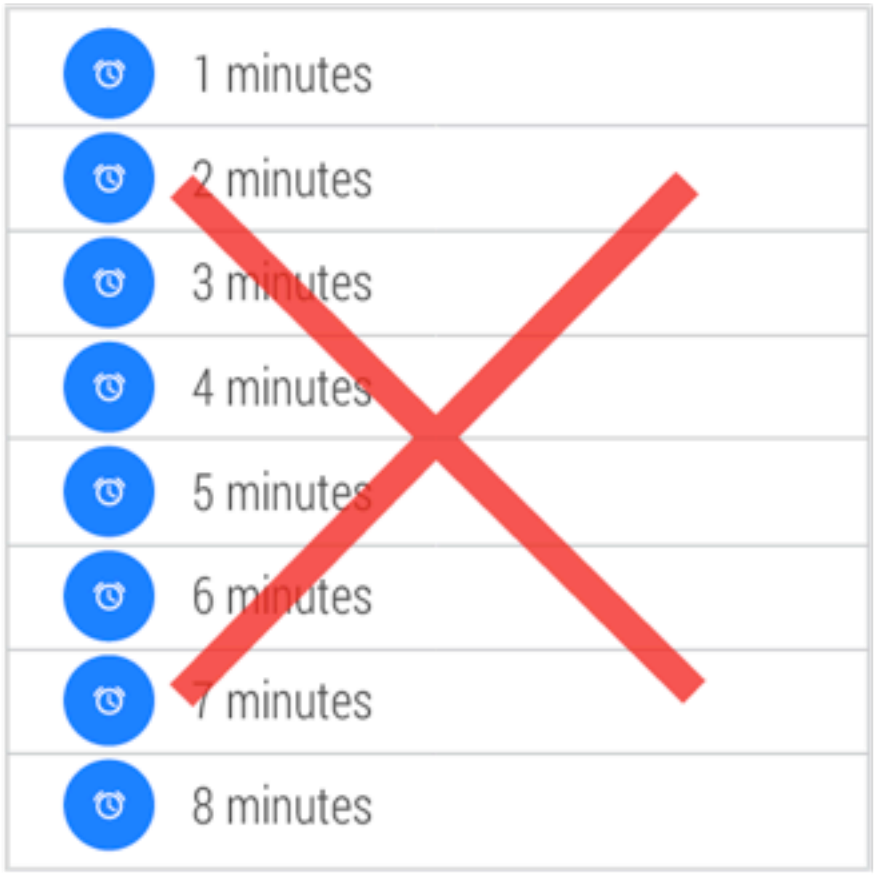
Do one thing, really fast

Don’t be a constant shoulder tapper

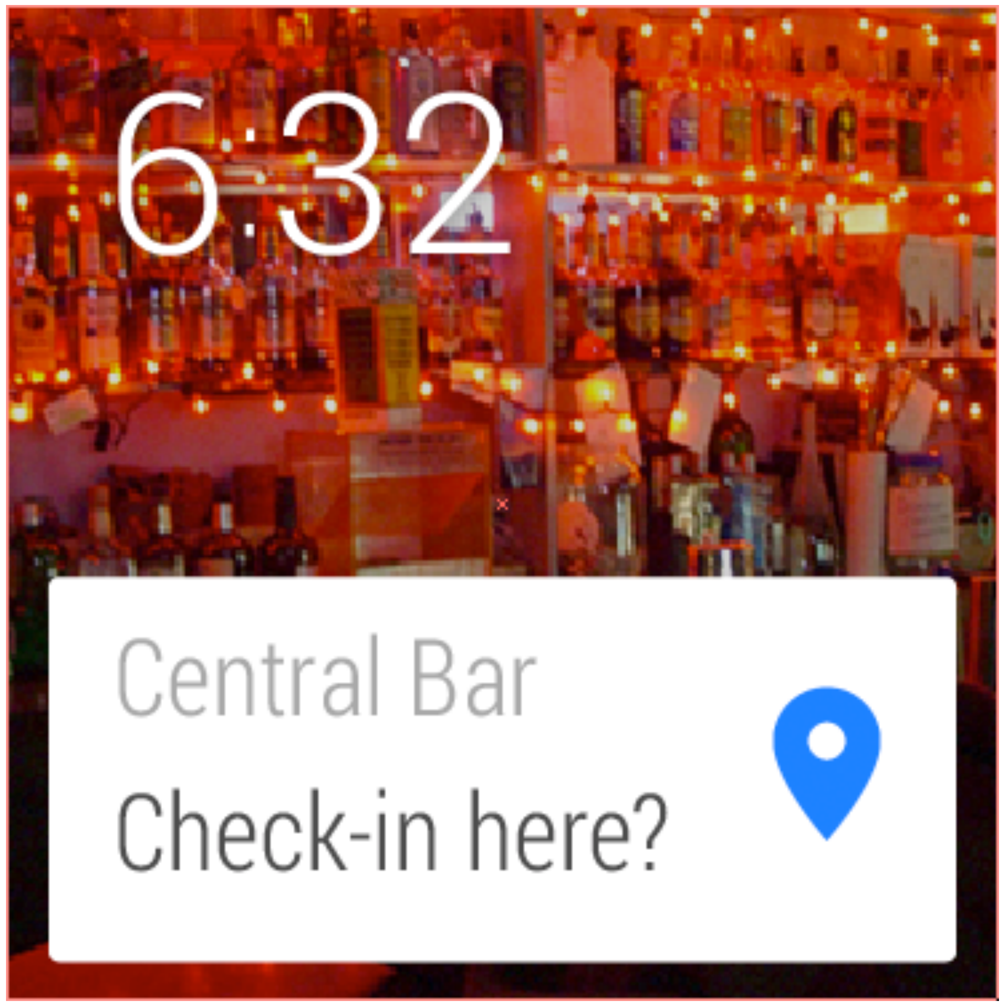
Design for the corner of the eye

Design for big gestures





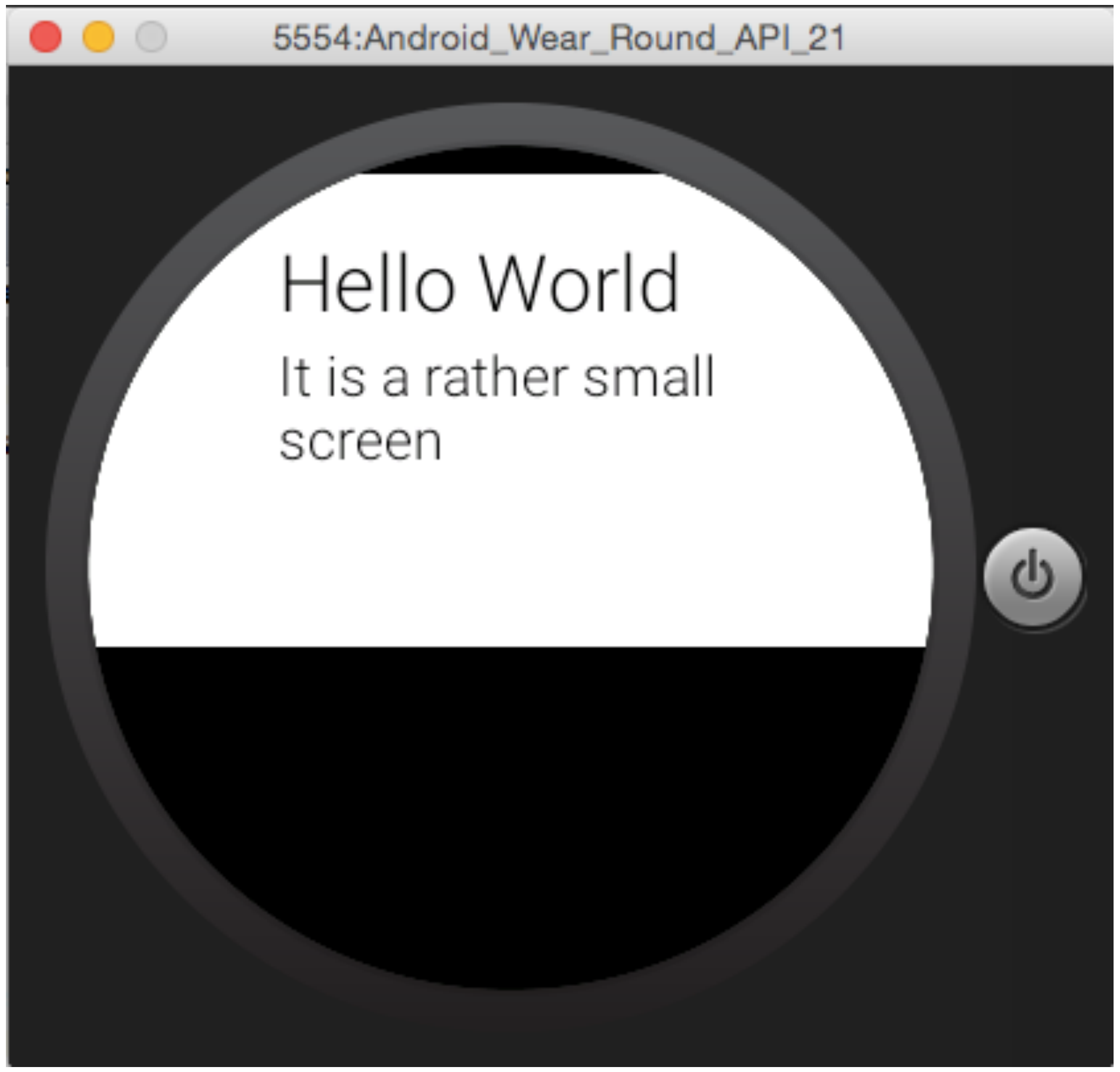
Think about stream cards first



Android Wear Layouts & Widgets

BoxInsetLayout
CardFragment
CircledImageViewConfirmationActivity
CrossFadeDrawable
DelayedConfirmationView
DismissOverlayView
DotsPageIndicator
GridViewPager
GridPagerAdapter
FragmentGridPagerAdapter
WatchViewStub
WearableListView

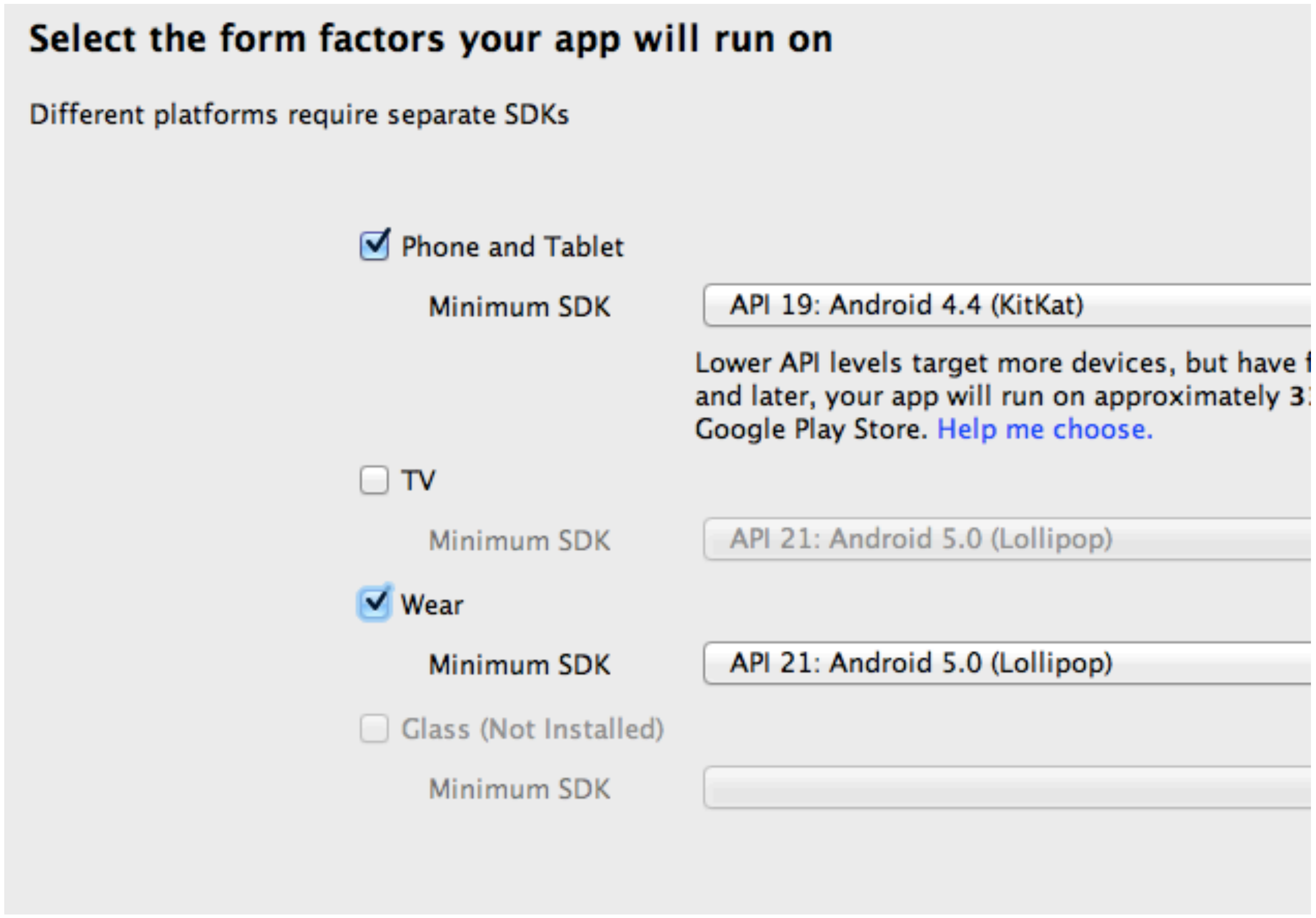
Wear App With Card



Hello World

It is a rather small screen

Android Studio

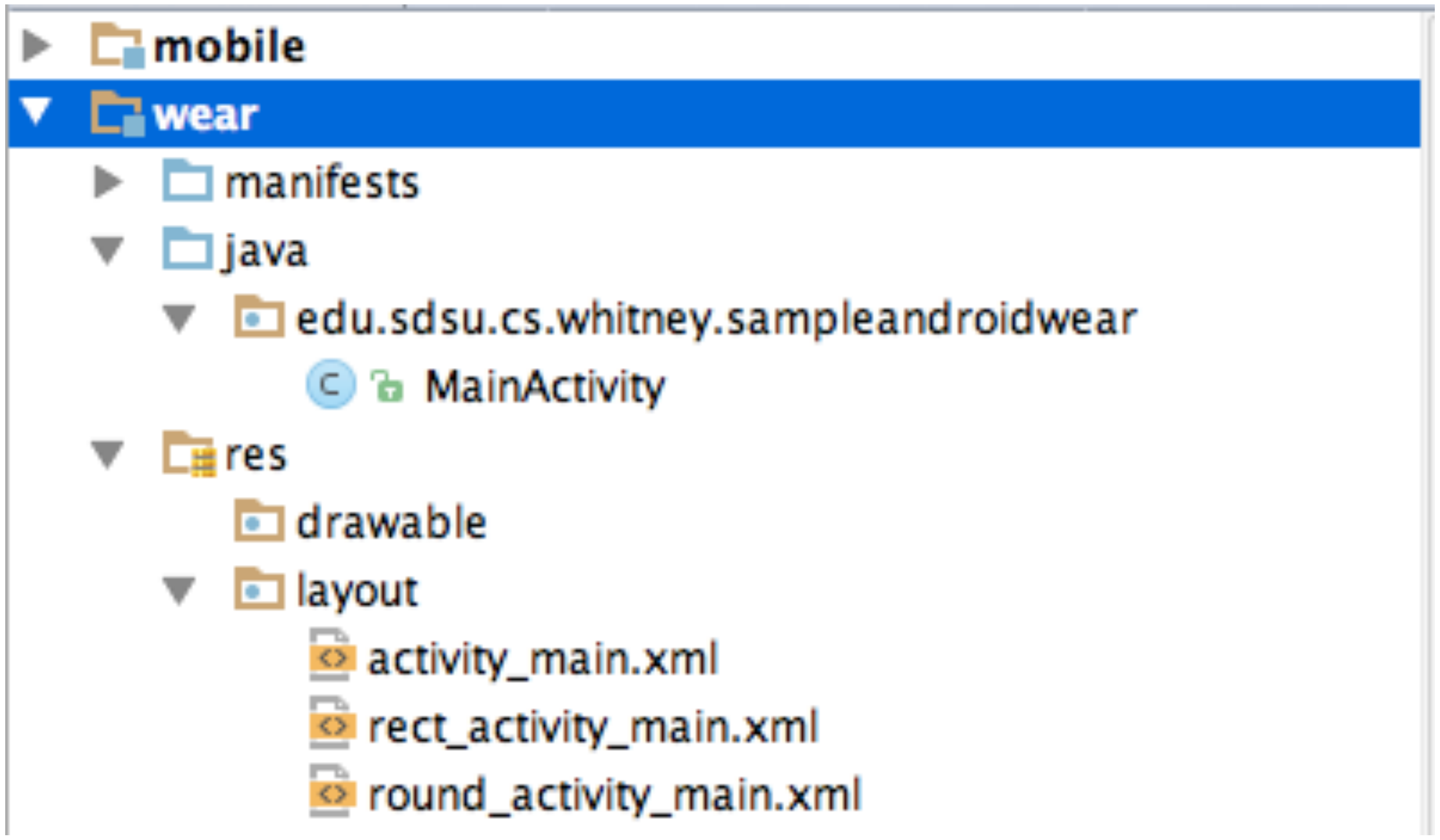


Select the form factors your app will run on

Different platforms require separate SDKs

- [✓] Phone and Tablet
  - Minimum SDK: API 19: Android 4.4 (KitKat)
  - Lower API levels target more devices, but have f and later, your app will run on approximately 3 Google Play Store. Help me choose.
- [ ] TV
  - Minimum SDK: API 21: Android 5.0 (Lollipop)
- [✓] Wear
  - Minimum SDK: API 21: Android 5.0 (Lollipop)
- [ ] Glass (Not Installed)
  - Minimum SDK:

You get Three layouts

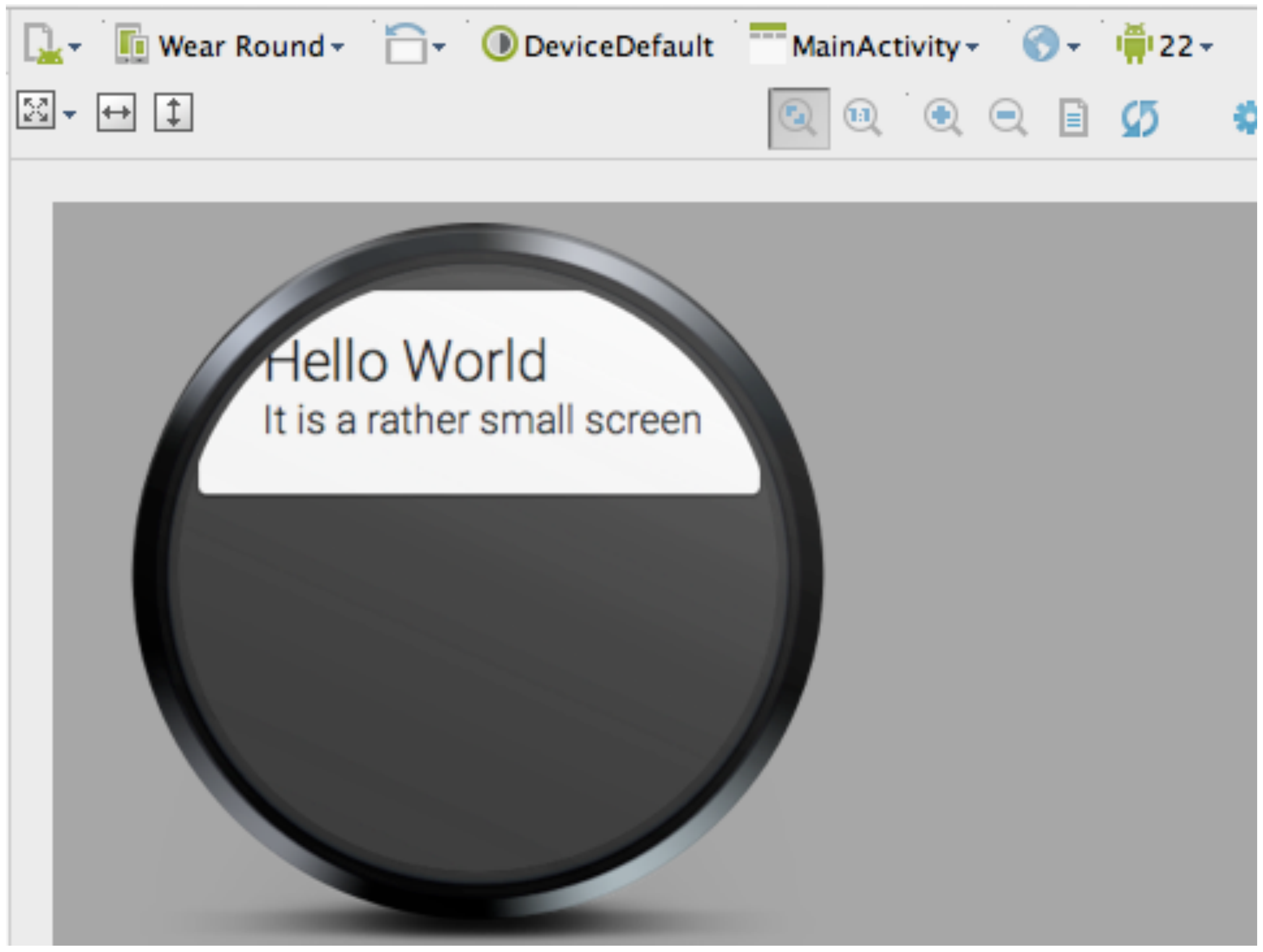

```
▶ mobile
▼ wear
    ▶ manifests
    ▼ java
        ▼ edu.sdsu.cs.whitney.sampleandroidwear
              MainActivity
    ▼ res
          drawable
        ▼ layout
              activity_main.xml
              rect_activity_main.xml
              round_activity_main.xml
```

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.wearable.view.BoxInsetLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_height="match_parent"
    android:layout_width="match_parent">

    <android.support.wearable.view.CardScrollView
        android:id="@+id/card_scroll_view"
        android:layout_height="match_parent"
        android:layout_width="match_parent"
        app:layout_box="bottom">

        <android.support.wearable.view.CardFrame
            android:layout_height="wrap_content"
            android:layout_width="fill_parent">

            <LinearLayout
                android:layout_height="wrap_content"
                android:layout_width="match_parent"
                android:orientation="vertical"
                android:paddingLeft="5dp">
                <TextView
                    android:fontFamily="sans-serif-light"
                    android:layout_height="wrap_content"
                    android:layout_width="match_parent"
                    android:text="Hello World"
                    android:textColor="@color/black"
                    android:textSize="20sp"/>
                <TextView
                    android:fontFamily="sans-serif-light"
                    android:layout_height="wrap_content"
                    android:layout_width="match_parent"
                    android:text="It is a rather small screen"
                    android:textColor="@color/black"
                    android:textSize="14sp"/>
```



Future Reading

Building Apps for Wearables

https://developer.android.com/wear/index.html

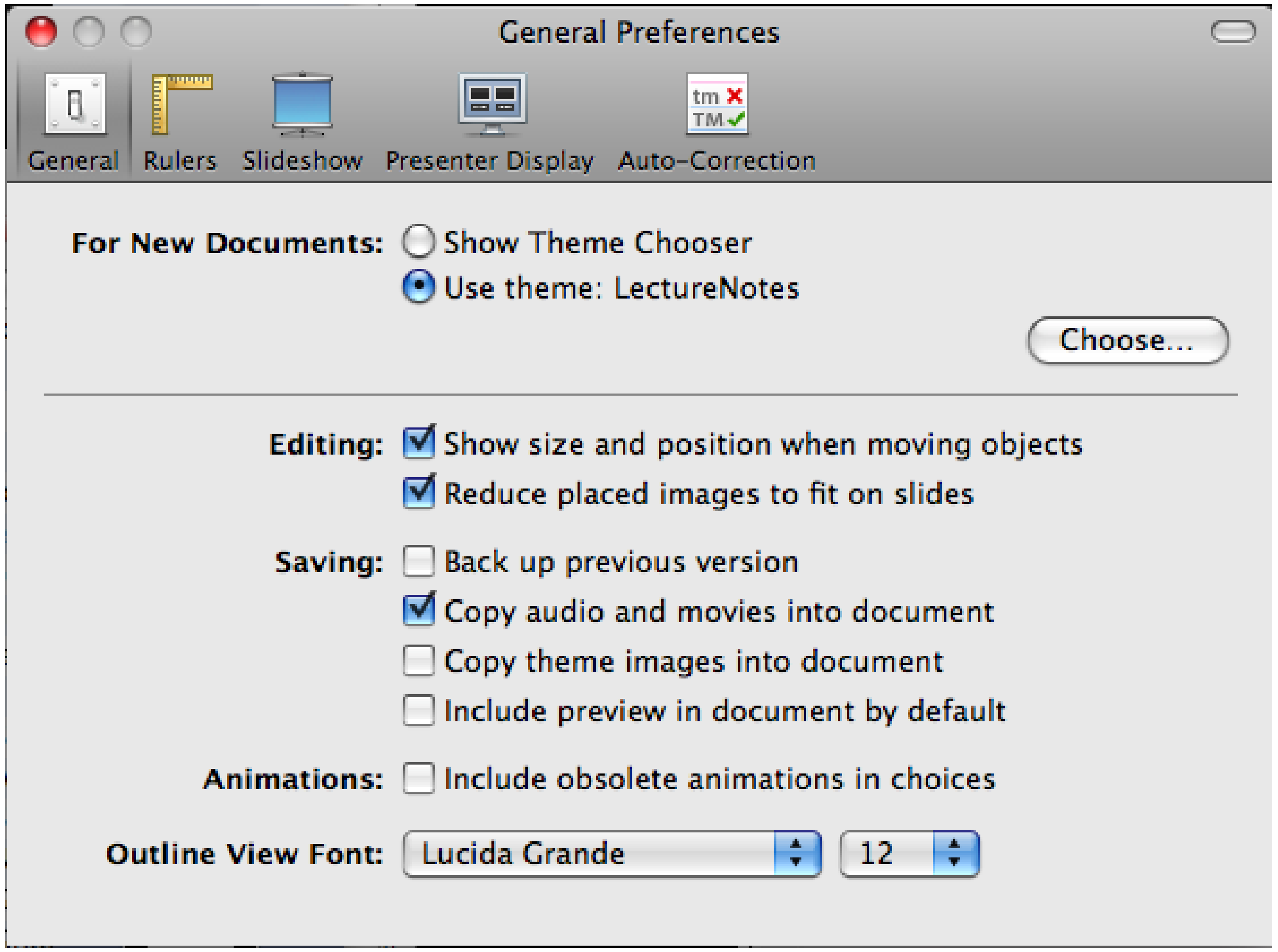
Design

https://developer.android.com/design/wear/index.html

User Interface Design For Programmers

A user interface is well-designed when the program behaves exactly how the user thought it would.

From http://www.apple.com/ilife/tutorials/#iphoto-intro



For New Documents: ( ) Show Theme Chooser [✓ Use theme: LectureNotes]

Editing: [✓ Show size and position when moving objects] [✓ Reduce placed images to fit on slides]

Saving: ☐ Back up previous version [✓ Copy audio and movies into document] ☐ Copy theme images into document ☐ Include preview in document by default

Animations: ☐ Include obsolete animations in choices

Outline View Font: Lucida Grande 12

Who is your user?

Patricia is an English professor who has written several well-received books of poetry. She has been using computers for word processing since 1980, although the only two programs she ever used are Nota Bene (an ancient academic word processor) and Microsoft Word. She doesn't want to spend time learning the theory of how the computer works, and she tends to store all her documents in whatever directory they would go in if you didn't know about directories.

http://www.joelonsoftware.com/uibook/fog0000000249.html

What does the user expect?

What is their mental model of the computer/application

Ask them

Perform usability studies

Use the Standards for you platform

It is what the users are used to

Six steps for designing good software

Invent some users

Figure out the important activities

Figure out the user model

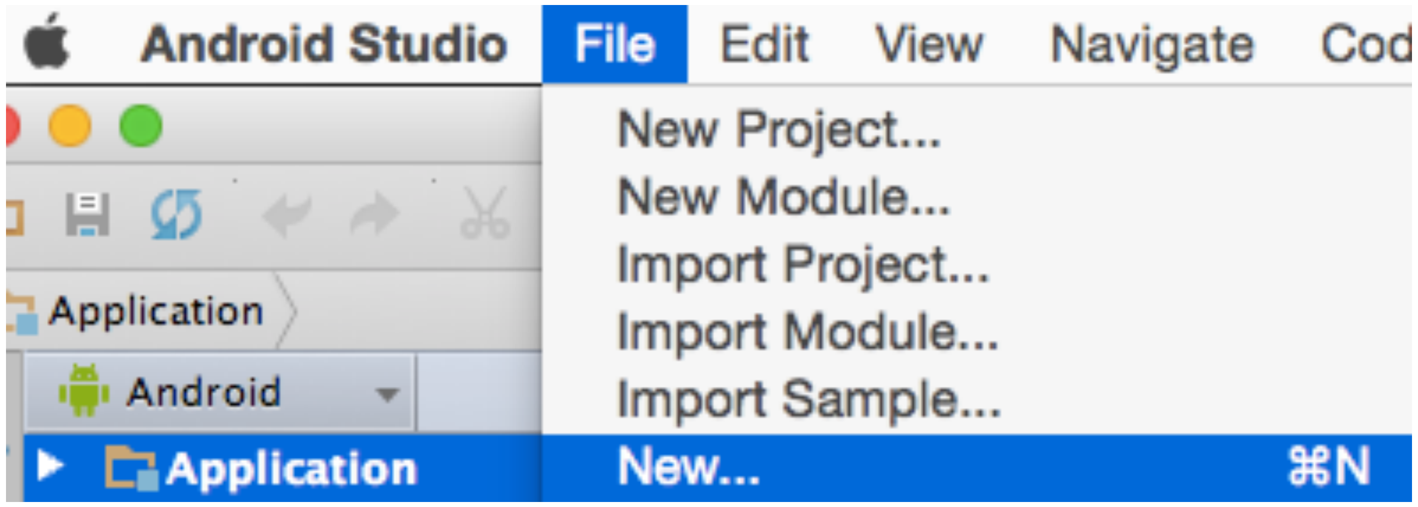
Sketch out the first draft of the design

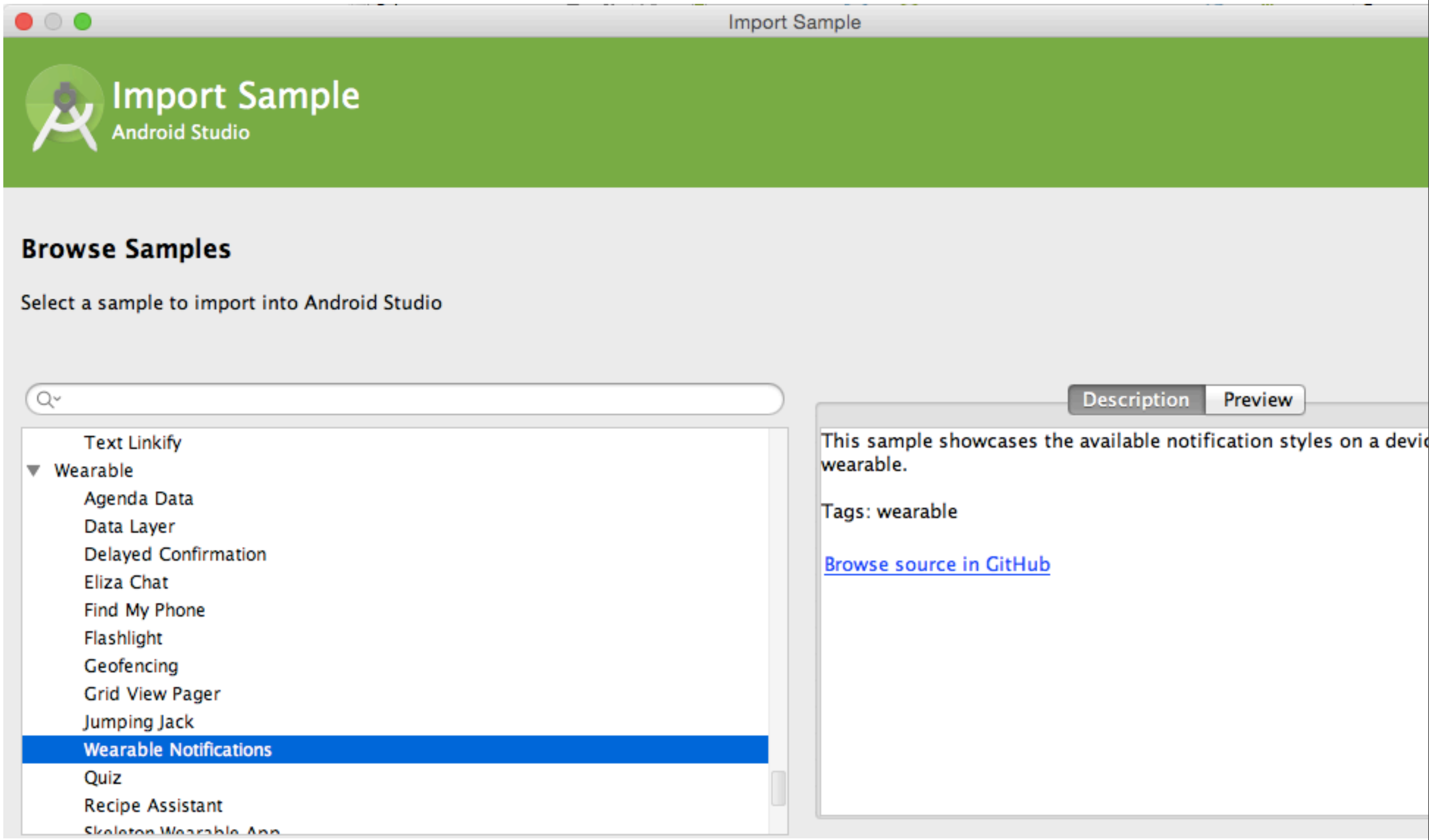
Iterate over your design again and again

Watch real humans trying to use your software.

One Last Thing

Android Studio - Sample Code





Browse Samples

Select a sample to import into Android Studio

- Text Linkify
- ▼ Wearable
  - Agenda Data
  - Data Layer
  - Delayed Confirmation
  - Eliza Chat
  - Find My Phone
  - Flashlight
  - Geofencing
  - Grid View Pager
  - Jumping Jack
  - **Wearable Notifications**
  - Quiz
  - Recipe Assistant

This sample showcases the available notification styles on a devic
wearable.

Tags: wearable

Browse source in GitHub

Tuesday, April 28, 15