Wallpaper, Widgets & Wear

Glanceable displays on Android

Elizabeth Mezias

September 2016

DevFest DC

Glanceable displays on Android

Elizabeth Mezias

Modo Labs
Senior Software Engineer

Sample code:

https://goo.gl/0Qo5kX

DevFest DC

Quantified Self

self knowledge through numbers

- ★ Smile detection & amplification
- ★ Tracking weight, matched to mood

  "…tracking emotions was changing them. The practice of "observing inward" with technology helped her notice, acknowledge and shape her emotional state."

FitBit - Swarovski Shine - UP2

- ★ Sensors for temp, heart rate, calories, and sleep
- ★ Best of 2016
  https://goo.gl/rQBB8M
- ★ Wide variety of styles and sensors
- ★ Great results when users can connect with each other

DevFest DC

What is wearable?

- ★ Activity detection via Fit, Awareness API
  - ○ Steps
  - ○ Minutes
- ★ Google Now
- ★ Always at hand

DevFest DC

Android Wear

- ★ Bridge your app easily with notifications
- ★ Be helpful & responsive, handle speech input
- ★ Be specific using time and location
- ★ Keep it simple & fast - short, relevant, immediate

Design thinking...

- ★ Build meaning with an app’s data to go beyond the obvious location and time inputs
- ★ Create new contexts, set a style
- ★ Infrequent interactions - do not pester your user
  - What can be measured?
  - Assign meaning from the data to colorful imagery

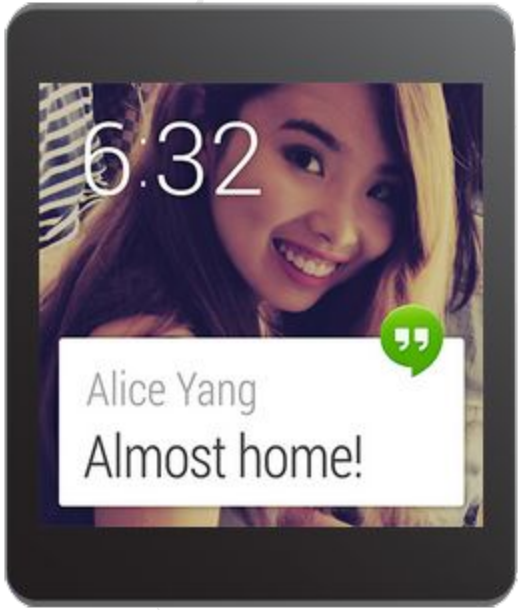
DevFest DC

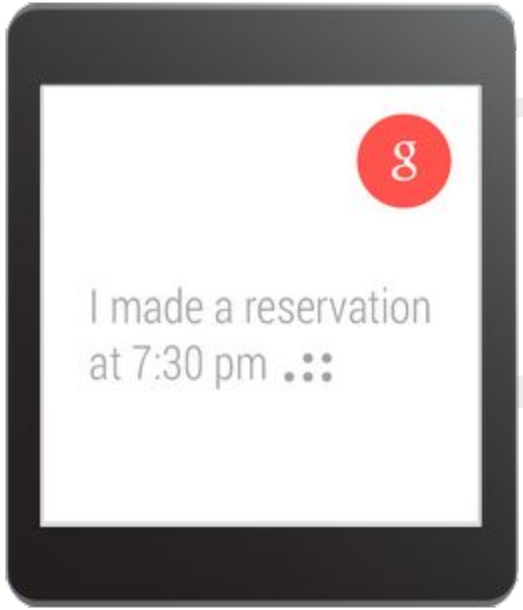
3 App Patterns on Wear

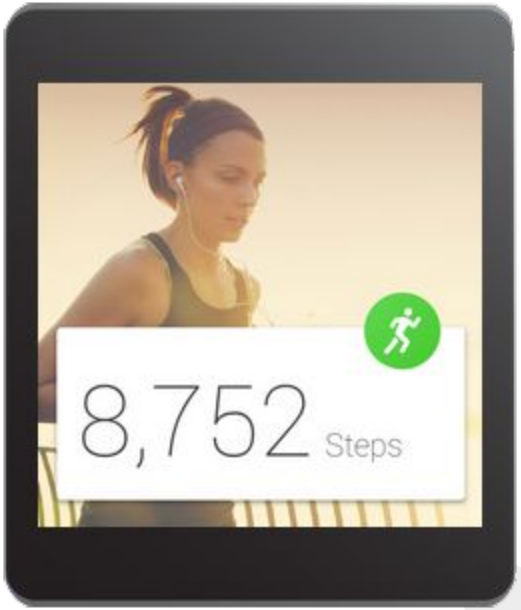
- ★ Watch Faces
- ★ Notifications & responses (speech, tap)
- ★ On Board Apps
  - Input Speech device
  - Read device sensors, and track data for a phone app and responsive web portal

DevFest DC

What can WEar do?



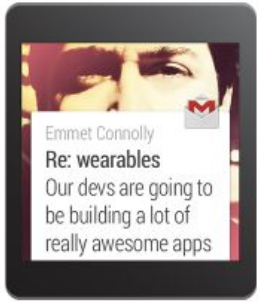




Where to? Wear 2
Now you have network access on board

NotificationCompat Builder

- ★ Set 3 Action Cards using WearableExtender
- ★ BigView style, 2 parts, peek and pull up
- ★ Hide the icon, set a background*
- ★ RemoteInput classes get speech back
- ★ Stack notifications -> setGroup creates a scroll
- ★ Drill down easily with addAction/setContentIntent



DevFest DC

Voice Actions

- ★ Define an intent filter
- ★ Handle the intent in your app
- ★ Update your app completion status

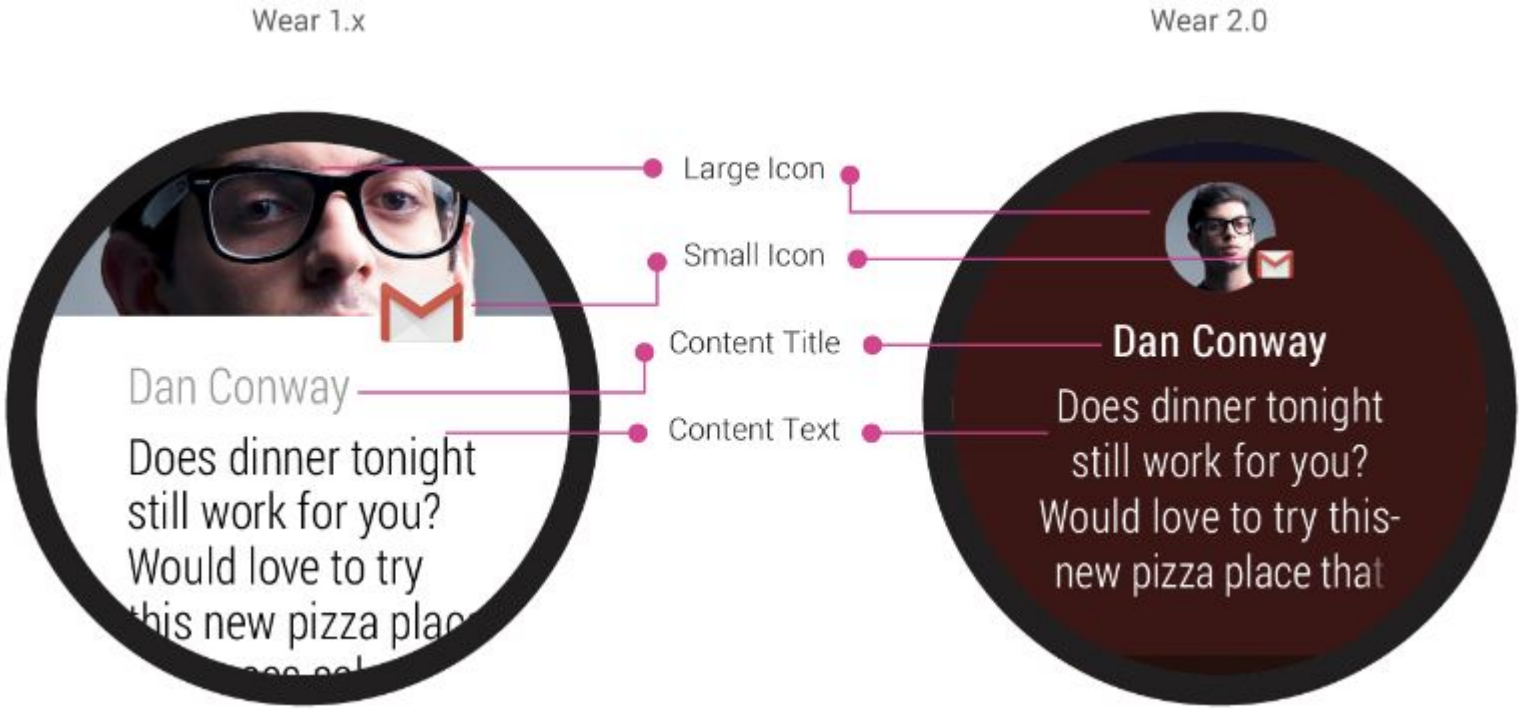
These are practical, common intents

Your app can answer them or call them

DevFest DC

New(ish) Notification Stuff

- ★ Heads up notification shows over anything on screen
- ★ Nougat can reply inside notifications
- ★ Media bar for notifications with playback to replace RV
- ★ New MessagingStyle for more customized notifications
- ★ Bundle notifications together (don’t pester)
- ★ Set lockscreen privacy for your users
- ★ Make your notifications show on Auto
- ★ Synchronize notifications - dismiss on both phone and watch (and Auto and Tablet)



Messaging Style + Smart Reply

d.google.com/voice-actions/

- Set alarm
- Set timer
- Initiate a phone call
- Start/stop a bike ride
- Start/stop a run
- Start/stop a workout
- Show heart rate
- Show step count
- Book a cab
- Play music from search
- Take a picture
- Record a video
- Open URL
- Open Application
- Take a note
- Search using a specific app

Voice example for Fit Actions

- ★ “OK Google...start my bike ride"
  “OK Google...stop cycling”
  "OK Google...track my run"
  "OK Google...stop workout"
- ★ Create your own

DevFest DC

Final Points

- ★ A new design and UX
- ★ No hands interactive, short time out
- ★ You can create gestures to start your app
- ★ Wear has limits: *no webkit, print, backup, widgets, or usb*

Home Screen Widgets

Live Widgets

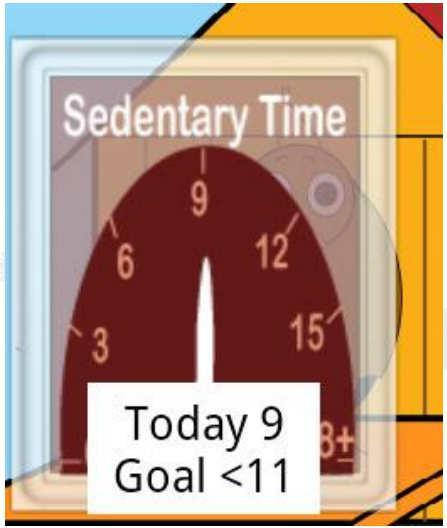
Collection Lists & Stacks, Shortcuts

DevFest DC

Meter Widgets

Stanford Medical Apps

- ★ Public Health
- ★ Accelerometer data
- ★ “MVPA” Minutes
- ★ “Sedentary” Hours
- ★ Resets @midnight
- ★ Always-on service



HOW-TO

Extend 2 classes and override 4 methods

1. RemoteViewsFactory - bind Collection data to widgets
    a. getViewAt() - inflate a layout
    b. onDataSetChanged() - show the collection
2. AppWidgetProvider - respond to the user, timed updates
    a. onReceive(), like any Broadcast Receiver
    b. onUpdate(), code runs on time, updatePeriodMillis

Add it to the Manifest

RECEIVER - AppWidgetProvider class like WeatherWidget

Set the proper intent filter on it ***APPWIDGET_UPDATE***

appwidget-provider xml file is required

SERVICE - RemoteViewsFactory is a service

Set ***BIND_REMOTEVIEW*** permission for the Android System

PROVIDER (optional) Declare a Content provider like the WeatherDataProvider class to provide data to the widget

Widget MetaData

Looks like ../res/xml/appwidget_provider.xml

AppWidgetProviderInfo attached to the intent filter

- ★ App Widget's initial layout
- ★ Update frequency
- ★ Width and Height
- ★ WIDGET_CATEGORY, homescreen and/or keyguard

1

```
<!-- The content provider serving the data -->
<provider android:name=".weatherwidget.WeatherDataProvider"
        android:authorities=".weatherwidget.provider" />
```

2

```
<!-- The widget provider -->
<receiver android:name=".weatherwidget.WeatherWidget”>
    <intent-filter>
    <action android:name
        ="android.appwidget.action.APPWIDGET_UPDATE" />
    </intent-filter>
    <!-- This specifies the widget provider info -->
    <meta-data android:name="android.appwidget.provider"
        android:resource="@xml/widgetinfo" /> </receiver>
```

3

```
<!-- The RemoteViews service -->
<service android:name=".weatherwidget.WeatherWidgetService”
    android:permission="android.permission.BIND_REMOTEVIEWS"/>
```

RemoteViewService & RemoteViewsFactory

- ★ Override OnCreate to initialize
- ★ Override onDataSetChanged to update the views
- ★ Clean up in OnDestroy, close cursors, close files
- ★ Examples: WeatherWidgetService, StackWidgetSvc
- ★ Pro Tip:

  Nest the Factory class to use the Service Context

```
public RemoteViews getViewAt(int position) {

    final RemoteViews rv =
        new RemoteViews(packageName,R.layout.item1);
    rv.setTextViewText(R.id.text, city);
    rv.setImageViewResource(R.id.image, image_id);
    rv.setOnClickFillInIntent(R.id.image,
        (new Intent()).putExtra(EXTRA_CITY_ID, city));
    return rv;
}
```

ACTION_APPWIDGET_UPDATE

- ★ Set views in the widget

  setRemoteAdapter(RemoteViewsService)

- ★ Bind action to the views

  SetPendingIntentTemplate(), view onClick action

- ★ Code an action on the collection

  SetOnClickPendingIntent(), touch list item

Req’d Code

```
@Override
public void onUpdate(Context ctx, AppWidgetManager
    widgetManager, int[] widgetIds) {
    for (int i = 0; i < appWidgetIds.length; ++i) {
        appWidgetManager.updateAppWidget(widgetIds[i], rv);
    } //required for loop
    super.onUpdate(ctx, widgetManager, widgetIds);
}
```

DevFest DC

RemoteViewsService

```
Intent i = new Intent...
i.putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID,
appWidgetIds[i]);
i.setData(Uri.parse(i.toUri(Intent.URI_INTENT_SCHEME)));
RemoteViews rv = new RemoteViews...
rv.setRemoteAdapter(R.id.stack_view, i);
rv.setEmptyView(R.id.stack_view, R.id.empty_stack_view);
```

Fill in the blank

```
Intent viewItem = new Intent(context,
ImageActivity.class);
rv.setPendingIntentTemplate(R.id.stack_view,
PendingIntent.getActivity(context, 0, viewItem,
PendingIntent.FLAG_UPDATE_CURRENT));
rv.setOnClickPendingIntent(R.id.image,
PendingIntent.getActivity(context, 0, viewItem,
PendingIntent.FLAG_UPDATE_CURRENT));
```

DevFest DC

Bonus Points, onReceive

```
public void onReceive(Context ctx, Intent intent) {
    final String action = intent.getAction();
    if(action.equals(REFRESH_ACTION)) {
        //Button pressed, update temperatures
    } else if(action.equals(CLICK_ACTION)) {
        //List item touched
    }
}//end onReceive
```

DevFest DC


GET DEMO

Wallpaper X
Live Wallpaper

More Glanceable Experiences
Part 3

DevFest DC

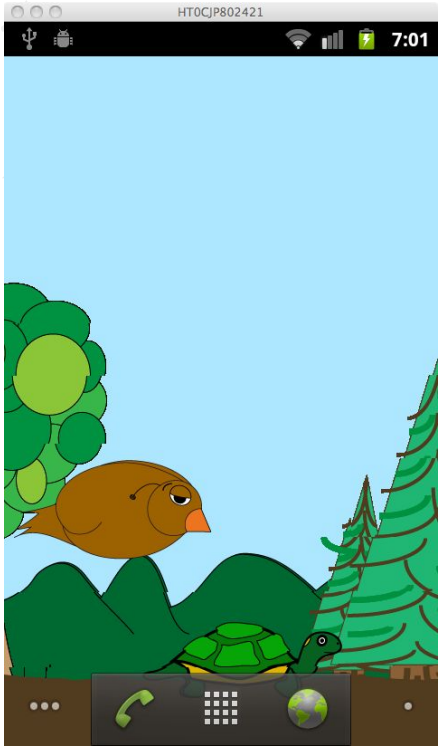
Intel & University of Washington

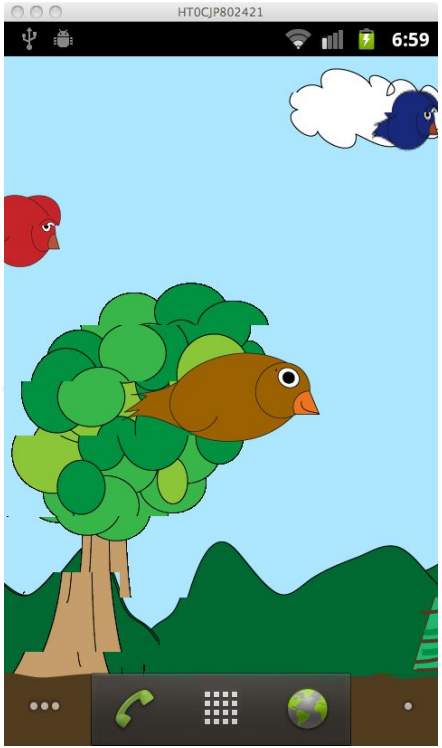
- ★ UbiFit
- ★ Low-cost sensing
- ★ Artificial Intelligence
- ★ Bio-feedback
- ★ Activity based
- ★ Ubiquitous computing

ShutEye, their latest...

DevFest DC

MILES (Stanford Medical School)





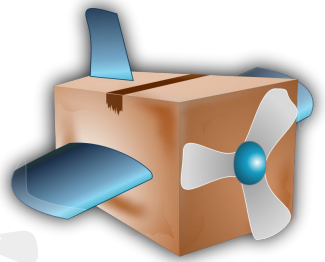
And the winner is...

- ★ Social, mConnect
- ★ All were a success
- ★ The work continues

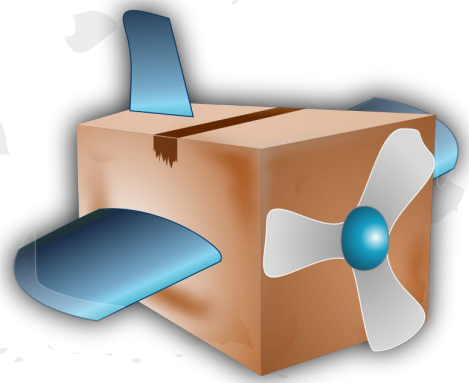
DevFest DC

HOW-TO

- ★ WallpaperService class, onCreateEngine( )
- ★ Engine logic - use a Handler and runnable
  - ○ draw to a Canvas
  - ○ Inflate a layout, call view.draw()
- ★ onOffsetsChanged( ) - don’t go there
- ★ onVisibilityChanged( ) - onSurfaceDestroyed( )
  start/stop the Handler, cleanup, exit
- ★ onTouchEvent( ) - user interaction (careful!)



```
public class MyWallpaperService extends WallpaperService
{
    @Override
    public Engine onCreateEngine() {
            return new MyWallpaperEngine();
     }
    class MyWallpaperEngine extends Engine {
        Handler handler = new Handler();
        Runnable drawRunner = new Runnable() {
            @Override
            public void run() {
                drawPaper();
            }
        }; //end drawRunner
```



DevFest DC

```
public MyWallpaperEngine() {
    //continues, initialize structures as needed
}

  @Override
public void onVisibilityChanged(boolean visible) {
    if(visible) {
        drawRunner.run();
    } else {
        handler.removeCallbacks(drawRunner);
    }
}

void drawPaper(){
    //drawing logic, 100 is 10 frames per second
    handler.postDelayed(drawRunner, 100);
}
```

```
@Override
public void onTouchEvent(MotionEvent event) {
    //touch adds a circle, reset clears the list
    super.onTouchEvent(event);
    if(touchEnabled  &&
        event.getAction() == MotionEvent.ACTION_UP) {
        int x = Float.valueOf(event.getX()).intValue();
        int y = Float.valueOf(event.getY()).intValue();
        touches.add(new TouchPoint(x, y));
        draw();
    }
}
```

DevFest DC

- Stanford University, MILES Study

  http://med.stanford.edu/miles
- MILES paper

  http://goo.gl/1ntS8n
- ShutEye, project site

  http://dub.washington.edu/projects/shuteye
- Mezcode, my portfolio site

  https://sites.google.com/site/mezcocorp/