

Developing For Android Wear



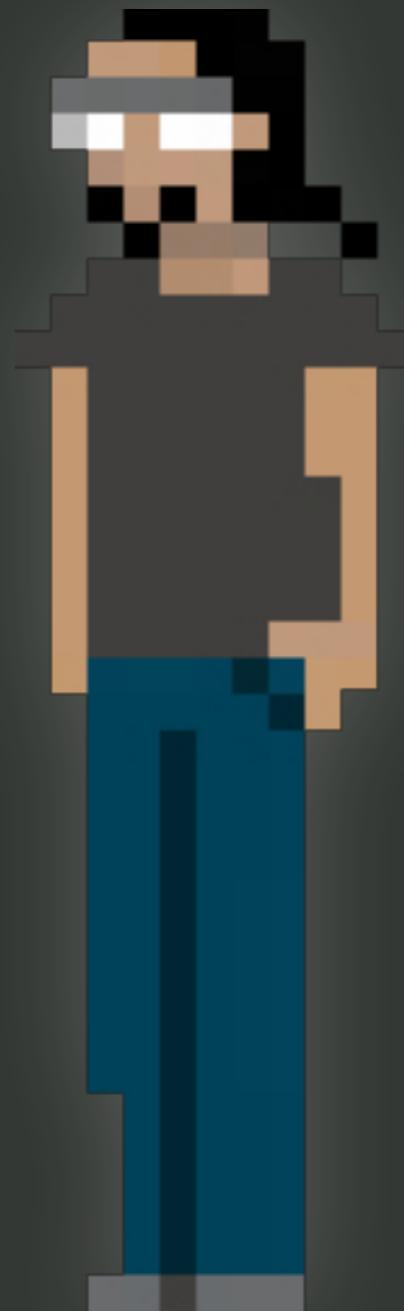
Welcome to the wrist world

Hi!

I'm Jérémie



@jeremie_laval



Agenda

What's Android Wear?

Designing for Wear

Coding for Wear

What's Android Wear?

What's Android Wear?

An Android-based platform
for every type of wearables

→ Smartwatches

Android Wear

Keeping you connected to what matters



ASUS ZenWatch

Smart companion and wellness manager, powered by Android Wear.



LG G Watch R

Classic watch design, only smarter.



Sony SmartWatch 3

Useful information when you need it, powered by Android Wear.



Moto 360

A modern timepiece powered by Android Wear.



Samsung Gear Live

Bright and stylish, the Samsung Gear Live moves with you.

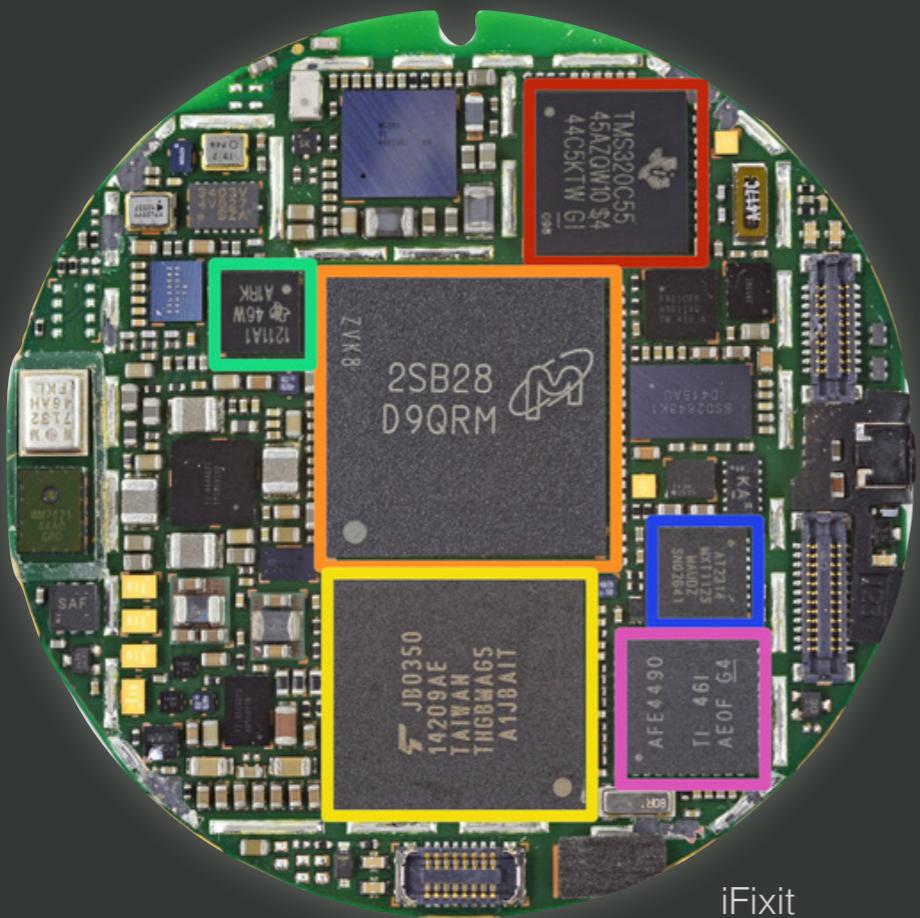


LG G Watch

Always with you, always-on. Powered by Android Wear.

Hardware

Ex: Moto 360



TI OMAP 3 SoC

1 GHz ARM CPU

512 MB RAM

→ Droid 2 (circa 2010)

Hardware

The Fun Stuff

- Bluetooth LE
- Microphone
- Pedometer
- Heart rate monitor (some)
- GPS (some)
- Accelerometer, gyroscope, barometer...

Software

- Android 5.0 Lollipop*
- Paired with an Android phone
- OTA updates pushed via phone app
- Wear apps are regular Android apps

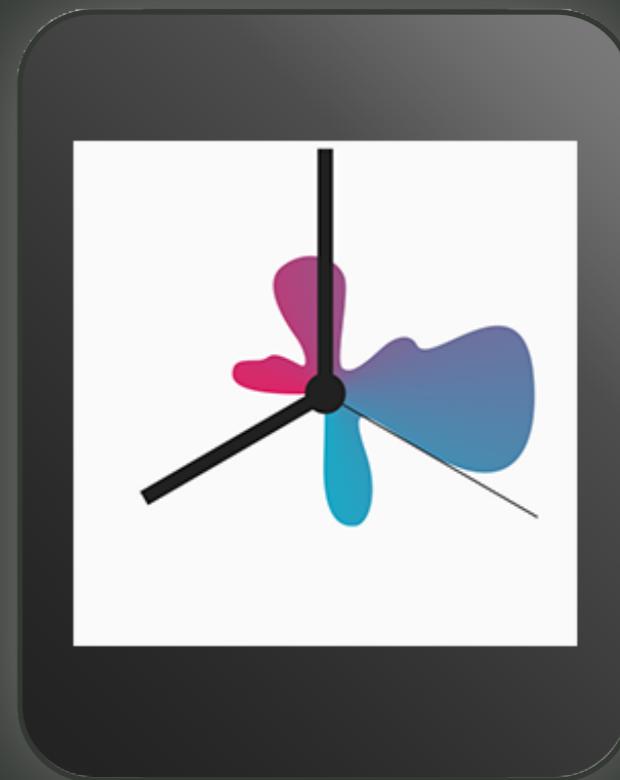
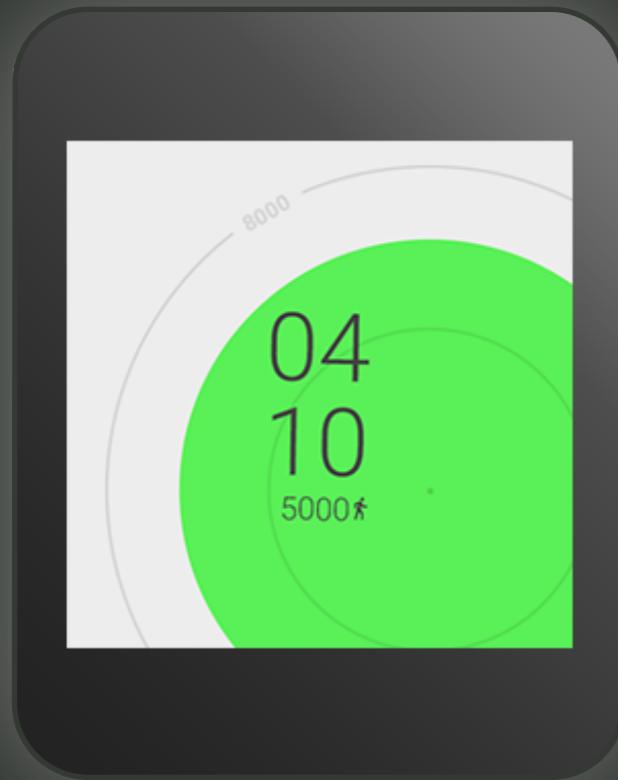
*minus some packages (webkit, print, backup, appwidget, ...)

Targeting Wear

Enhanced Notifications



Custom Watchfaces



Google Fit Apps



Full-blown Apps



Designing for Wear

User Experience

The right information

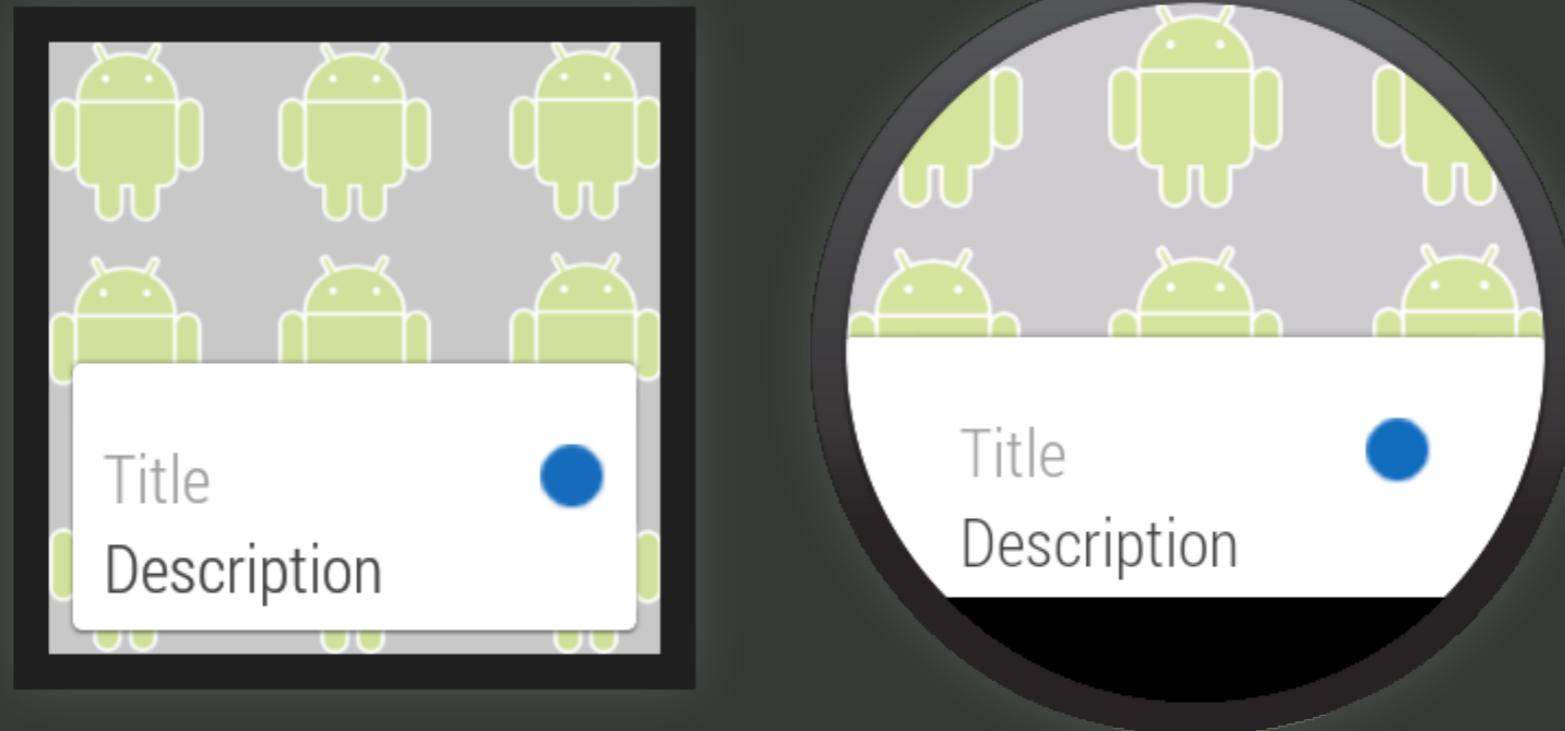
↳ at the right time

↳ at the right place

→ Context is king

Design Challenges

Screens



Size

280-320 pixels

Density

hdpi

Interactions

Touch

Un-precise

Gestures favored

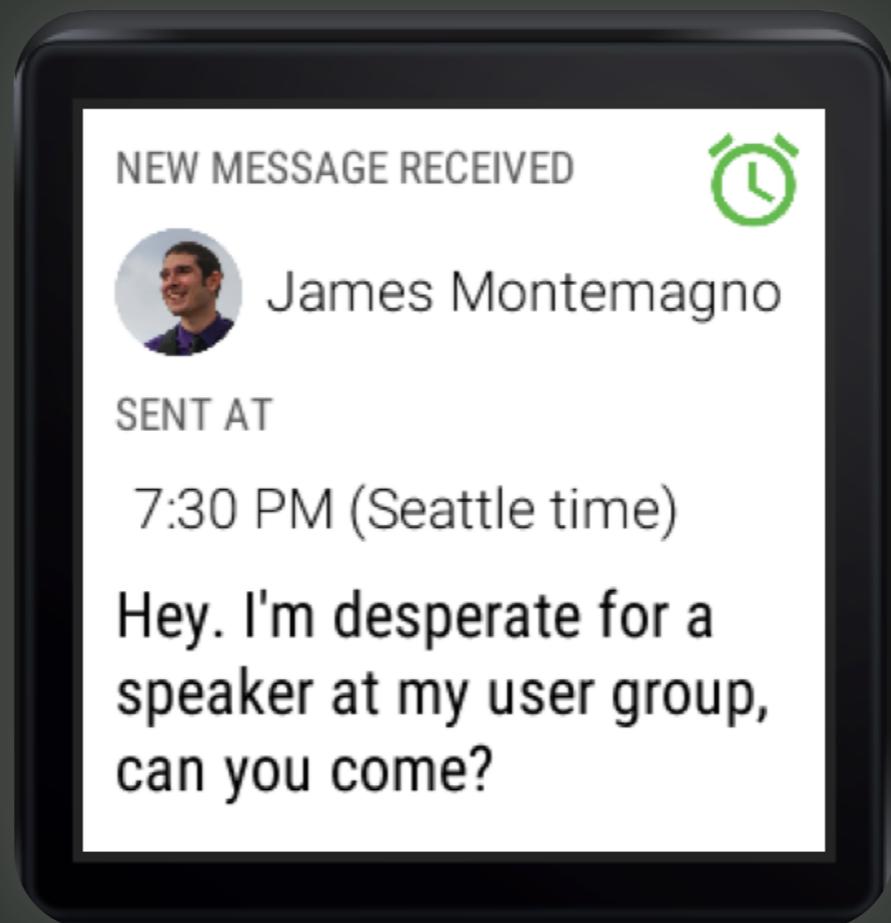
Sound

Input only

Commands

Dictation

Glanceable Design



Glanceable Design

Usage on-the-go

Random movements

Hands unavailable

Vision is vital

Micro Interactions

5s attention span

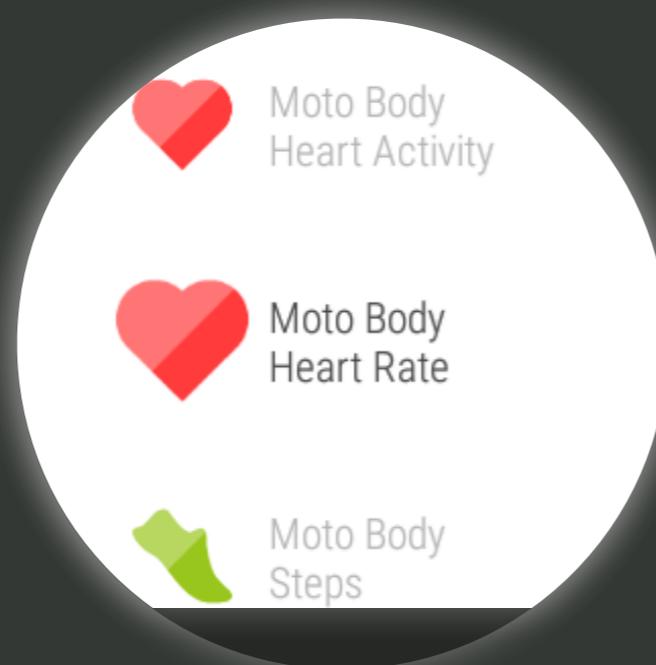
Self-contained task

App Navigation

- No accessible backstack
- Touch is tedious
- Swipe already a dominant gesture
- Probably too much

App Navigation

Multiple activities =
App drawer as your top navigation



→ Discoverable and voice triggerable

A Bad Wear App

Too much information

Too much functionality

Too much list/hierarchy

→ Don't shrink it, break it down

Coding for Wear

Moyeu



<https://github.com/garuma/Moyeu/>

Coding for Wear



It's just Android

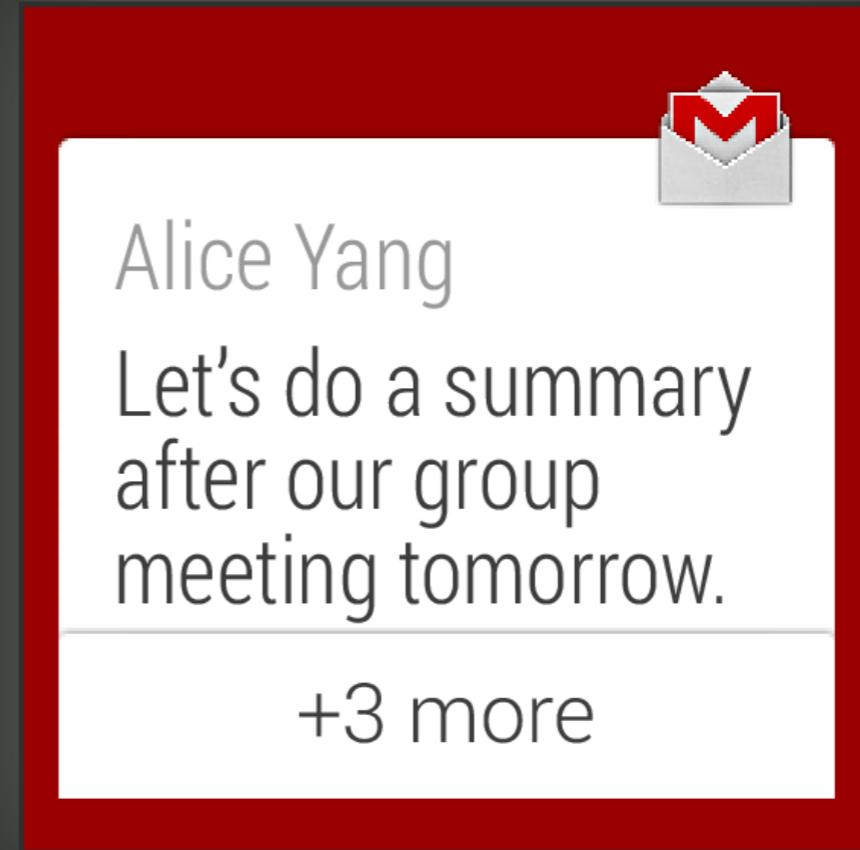
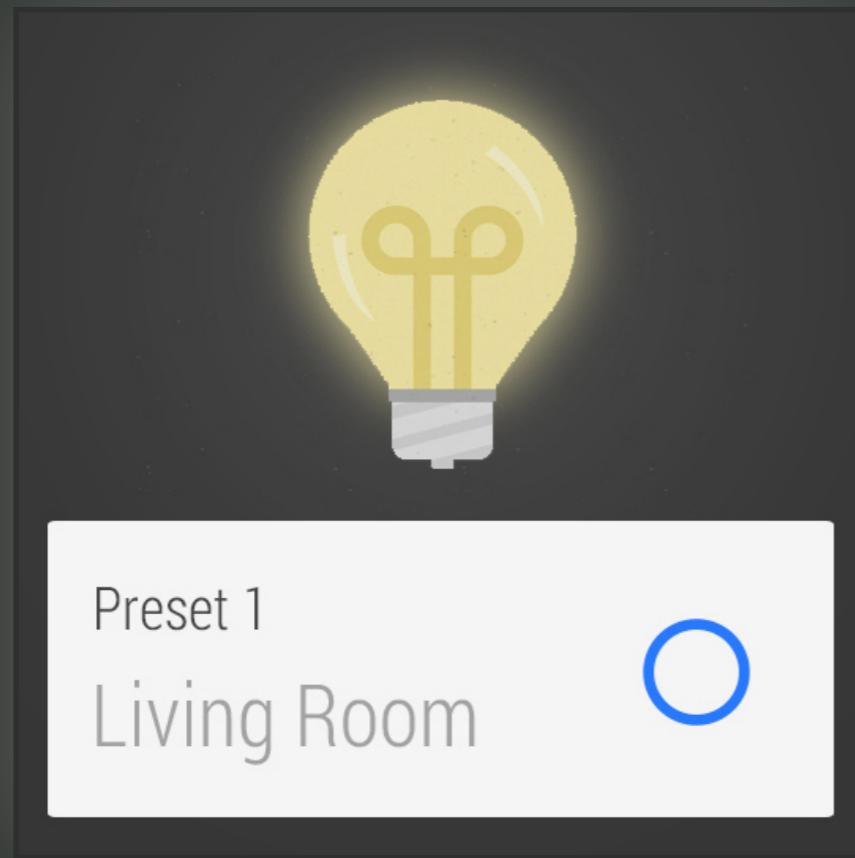
Wearable UI toolkit

An extra set of UI helpers
optimized for Android Wear



Xamarin.Android.Wear

Cards



Cards

CardFragment



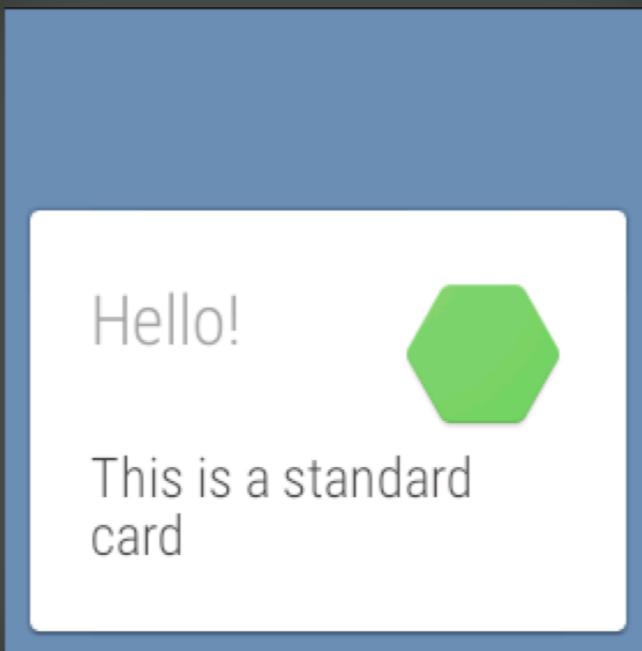
CardScrollView



CardFrame

Standard Cards

```
var cardFragment = CardFragment.Create ("Hello!",  
                                         "This is a standard card",  
                                         Resource.Drawable.Icon);  
  
FragmentManager.BeginTransaction ()  
    .Add (Resource.Id.frame, cardFragment, "card")  
    .Commit ();
```



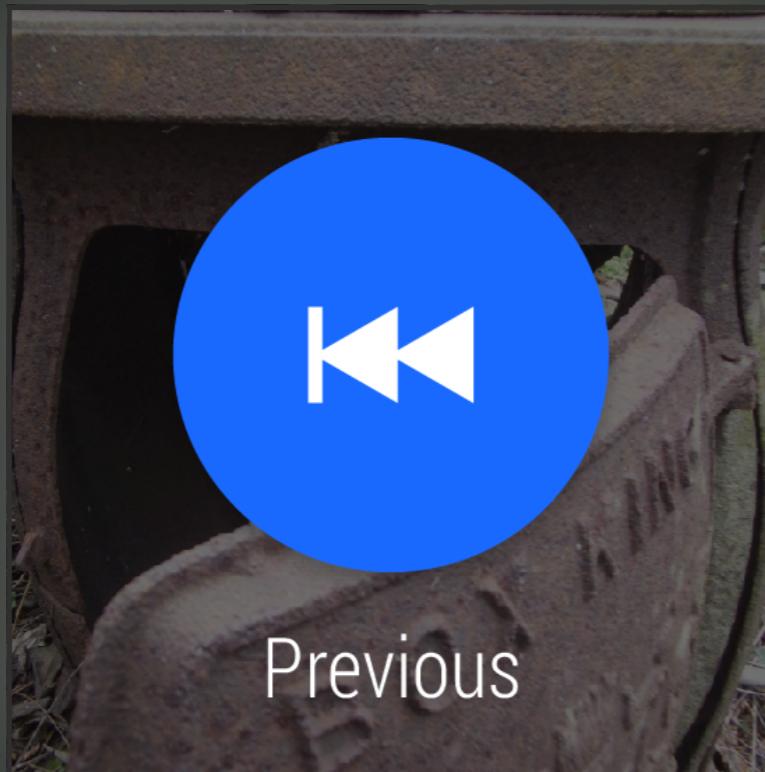
Custom Cards

```
public class StationCardFragment : CardFragment
{
    public override void OnCreate (Bundle savedInstanceState)
    {
        base.OnCreate (savedInstanceState);
        SetExpansionEnabled (false);
        SetCardGravity ((int)GravityFlags.Bottom);
    }

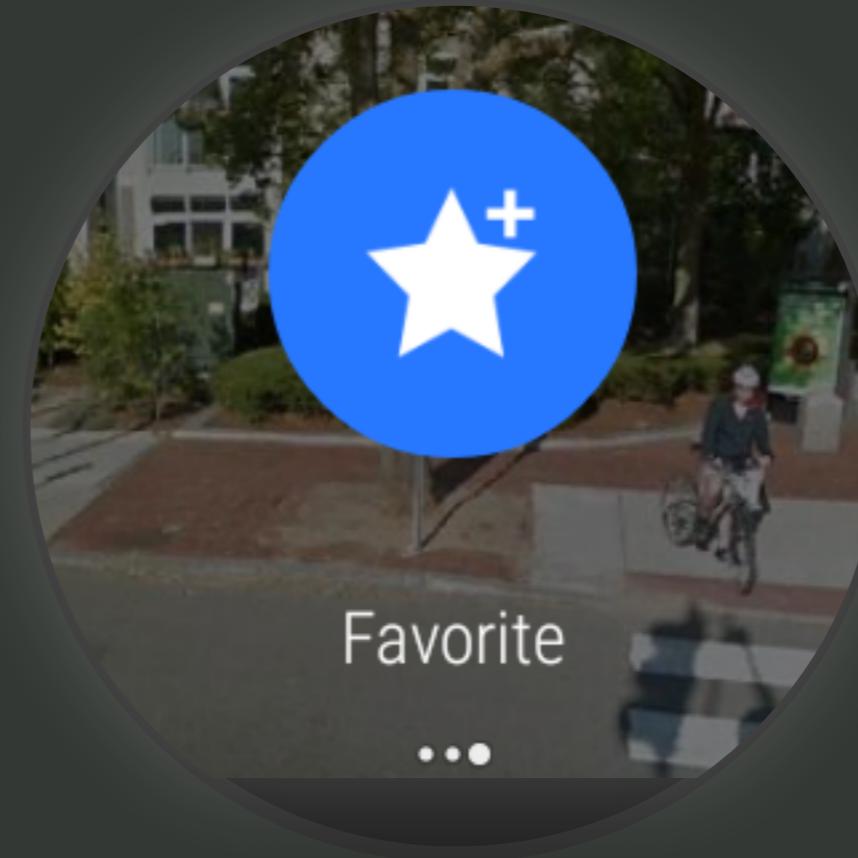
    protected override View OnCreateContentView (LayoutInflater inflater,
                                                ViewGroup container,
                                                Bundle state)
    {
        return inflater.Inflate (Resource.Layout.StationCardLayout, container, false);
    }
}
```



Action Buttons



Previous



Favorite

...



/garuma/Moyeu/blob/master/MoyeuWear/ActionButtonFragment.cs

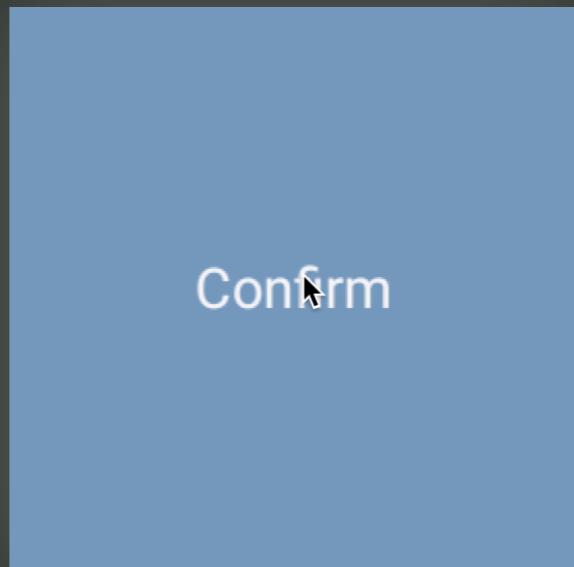
Standard Animations

AndroidManifest.xml

```
<activity android:name="android.support.wearable.activity.ConfirmationActivity" />
```

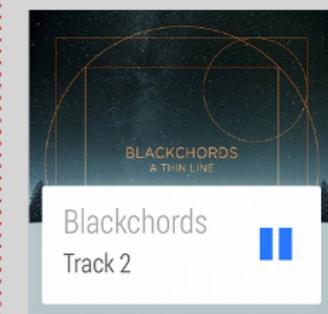
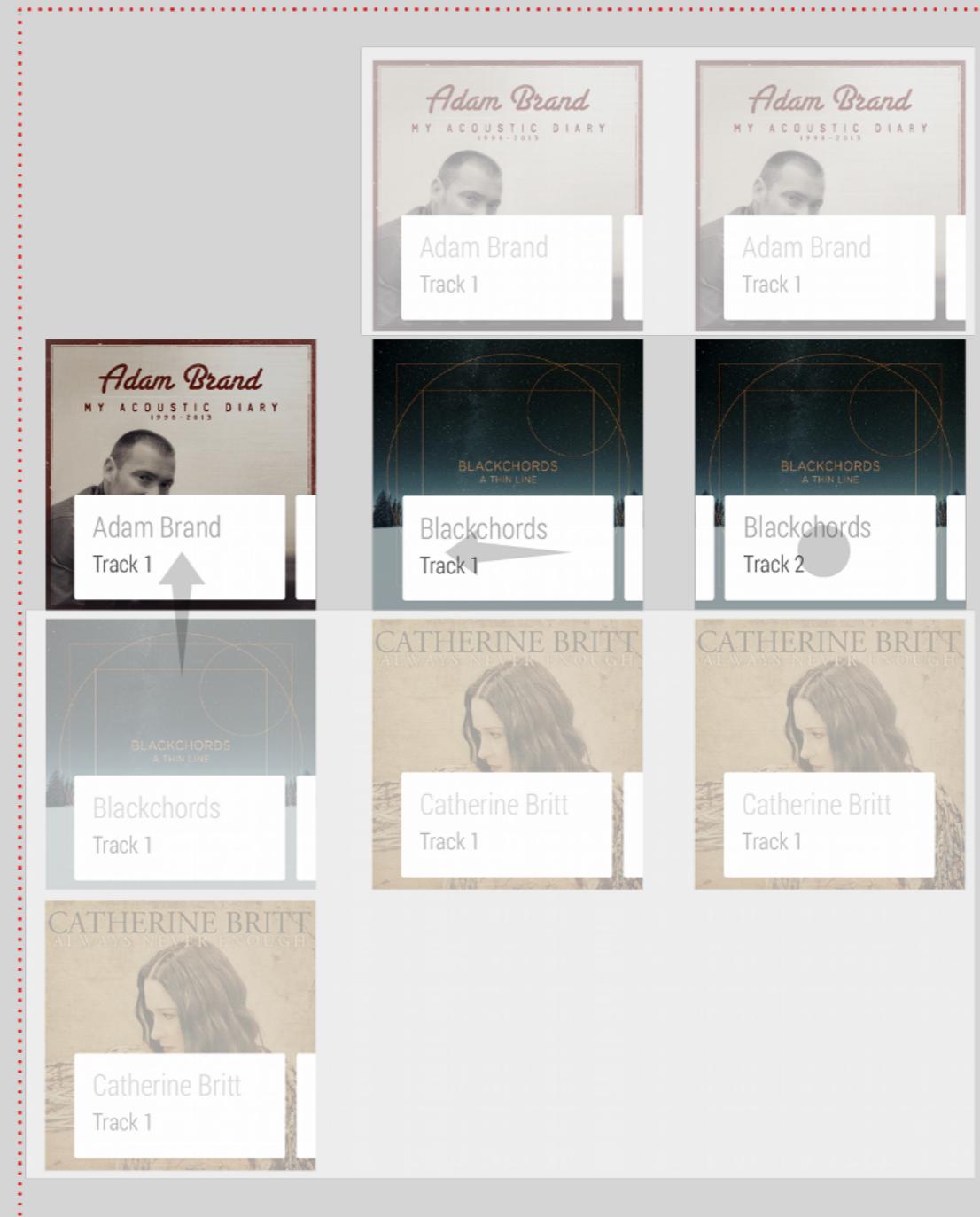
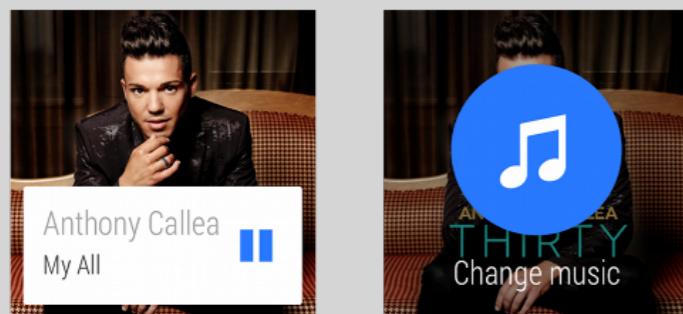
Code

```
var intent = new Intent (this, typeof(CheckmarkActivity));
intent.PutExtra (ConfirmationActivity.ExtraAnimationType,
                ConfirmationActivity.SuccessAnimation);
intent.PutExtra (ConfirmationActivity.ExtraMessage, "Great Success!");
StartActivity (intent);
```



2D Picker

● The Picker



2D Picker

Layout

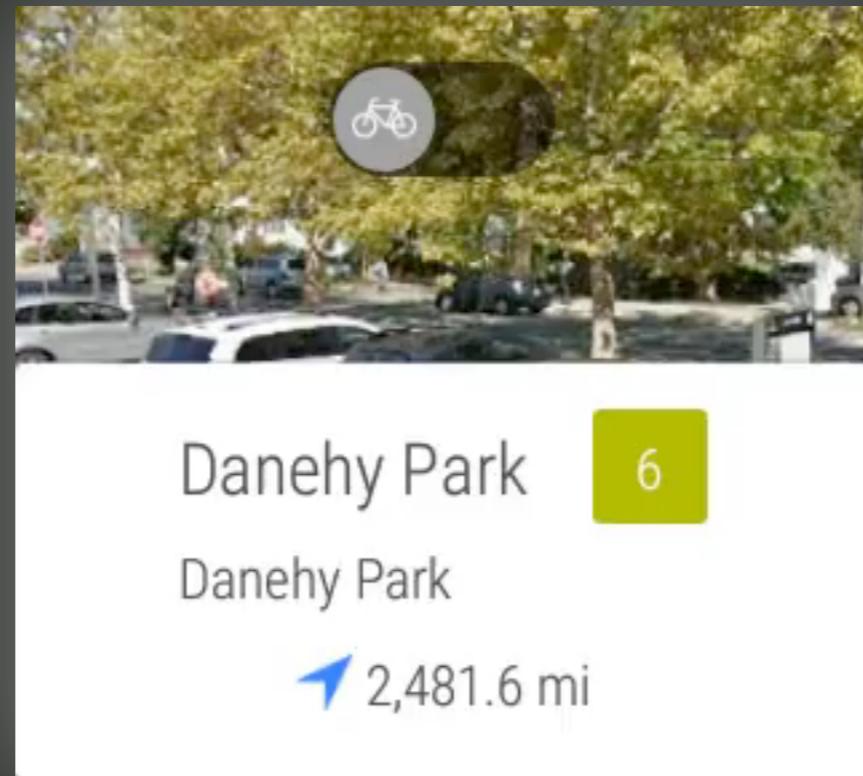
```
<android.support.wearable.view.GridViewPager  
    android:layout_width="fill_parent" android:layout_height="fill_parent"/>
```

Code

```
public class StationGridAdapter : FragmentGridPagerAdapter  
{  
    public override int RowCount { get { ... } }  
  
    public override Drawable GetBackgroundForRow (int row) ...;  
  
    public override int GetColumnCount (int row) ...;  
  
    public override Fragment GetFragment (int row, int column) ...;  
}
```



2D Picker



More Widgets

WatchViewStub

WearableListView

DotPageIndicator

DelayedConfirmationView

Voice Triggers

```
[Activity (Label = "Shopping", MainLauncher = true, ...)]  
public class MainActivity : Activity  
{
```

“Ok Google, Start <label>”

“Ok Google, start shopping”



Voice triggers

```
[IntentFilter (new[] { "vnd.google.fitness.TRACK" },
    DataMimeType = "vnd.google.fitness.activity/biking",
    Categories = new string[] { "android.intent.category.DEFAULT" })]
public class MainActivity : Activity
```

```
BikeActionStatus ActionStatus {
    get {
        return Intent.GetStringExtra ("actionStatus") == "CompletedActionStatus" ?
            BikeActionStatus.Stop : BikeActionStatus.Start;
    }
}
```

Full list: developer.android.com/training/wearables/apps/voice.html#SystemProvided



Communication



= Battery Consuming + Slow

~ % adb pull /sdcard/foo .

10 KB/s (17539200 bytes in 1597.854s)



17M



26m

Communication

Forget HttpClient



Communication Library



Google Play Services

(phone)



Xamarin.Android.Wear

(watch)

2 APIs

Message API

Low-level path-based messages

Data API

Phone ↔ Watch Dropbox

Setup

```
var client = new GoogleApiClientBuilder (this, this, this)
    .AddApi (WearableClass.Api)
    ...
    .Build ();
```



/garuma/Moyeu/blob/master/Moyeu/WearService.cs



/garuma/Moyeu/blob/master/MoyeuWear/MainActivity.cs

Node API

```
void GetStations ()  
{  
    WearableClass.NodeApi.GetConnectedNodes (client)  
        .SetResultCallback (this);  
}  
  
public void OnResult (Java.Lang.Object result)  
{  
    var apiResult = result.JavaCast<INodeApiGetConnectedNodesResult> ();  
    var nodes = apiResult.Nodes;  
    phoneNode = nodes.FirstOrDefault ();  
}
```

Peer to peer mesh network



Message API

```
var path = "/moyeu/Action/Favorite/";  
path += stationId + "?" + (cked ? "add" : "remove");  
WearableClass.MessageApi.SendMessage (client,  
                                      phoneNode.Id,  
                                      path,  
                                      new byte[0]);
```

- Targeted fire-and-forget
- Path-based
- Small payloads



Data API

Putting stuff in

```
var request = PutDataMapRequest.Create (SearchStationPath + "/Answer");
var map = request.DataMap;

var stationMap = new List<DataMap> ();
foreach (var station in nearestStations) {
    var itemMap = new DataMap ();
    itemMap.PutInt ("Id", station.Id);
    var asset = await CreateWearAssetFrom (station);
    itemMap.PutAsset ("Background", asset);
    itemMap.PutString ("Primary", primary);
    itemMap.PutDouble ("Distance", distance);

    stationMap.Add (itemMap);
}
map.PutDataMapArrayList ("Stations", stationMap);
map.PutLong ("UpdatedAt", DateTime.UtcNow.Ticks);

WearableClass.DataApi.PutDataItem (client, request.AsPutDataRequest ());
```



Data API

Getting it back out

```
public void OnDataChanged (DataEventBuffer dataEvents)
{
    var dataEvent = Enumerable.Range (0, dataEvents.Count)
        .Select (i => dataEvents.Get (i).JavaCast<IDataEvent> ())
        .First (de => de.Type == DataEvent.TypeChanged
                && de.DataItem.Uri.Path == SearchStationPath + "/Answer");
    var dataMapItem = DataMapItem.FromDataItem (dataEvent.DataItem);
    var map = dataMapItem.DataMap;
    var stations = new List<SimpleStation> ();
    var data = map.GetDataMapArrayList ("Stations");
    foreach (var d in data) {
        stations.Add (new SimpleStation {
            Id = d.GetInt ("Id", 0),
            Primary = d.GetString ("Primary", "<no name>"),
            Background = GetBitmapForAsset (d.GetAsset ("Background")),
            Distance = d.GetDouble ("Distance", 0),
        });
    }
}
```



Receiving Messages

```
WearableClass.DataApi.AddListener (client, this);
```

```
[Service]  
[IntentFilter (new[] { "com.google.android.gms.wearable.BIND_LISTENER" })]  
public class WearService : WearableListenerService  
{
```

- Passive helper
- One Wear service allowed
- Instantiated on-demand by Wear
- Messages automatically forwarded



Useful stuff

- blog.xamarin.com/tips-for-your-first-android-wear-app/
- developer.xamarin.com/guides/android/wear/
- developer.xamarin.com/samples/android/Android%20Wear/
- github.com/garuma/Moyeu
- github.com/garuma/xamarin-store-app/tree/wear-support
- <https://github.com/jamesmontemagno/DaysUntilXmas>
- www.google.com/events/io/io14videos