Android Wear programozás

Nyitrai István

nyitrai.istvan@bmeautsoft.hu

AutSoft BME

Amiről szó lesz

- A platformról dióhéjban
- Felületi újdonságok
- Fejlesztői környezet beállítása
- Értesítések
- Példa #1
- Kommunikáció
- Példa #2
- Példa #3
- (Példa #4)

Android Wear

A platformról dióhéjban

A platformról dióhéjban

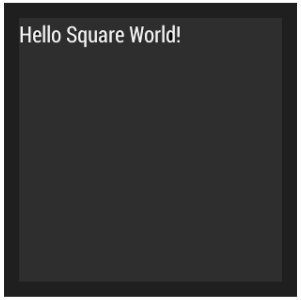
- A mobil eszköz kiterjesztése
- Hangutasítások
- Különálló alkalmazások (Activity, service, listeners)
  - Relatív gazdag API
- Szinkronizált adatok
  - Többféle üzenetváltási lehetőség az óra és a készülék között
- Egyszerű értesítések, egyszerű kezelőszervek, egyszerű felületek
- Szinkronizált értesítések
- Egy képernyő, lehetőleg egy akció
- (Pozíció meghatározása)
- Bluetooth LE

Android Wear

Felületi újdonságok

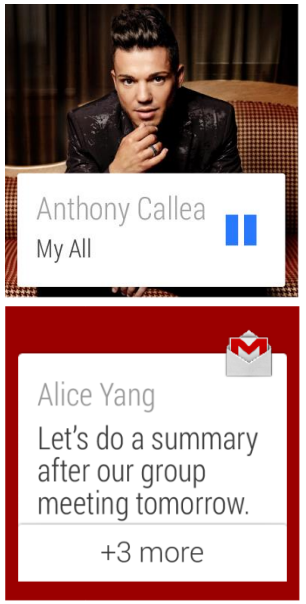
Felületi újdonságok

- WatchViewStub
  - Külön kerek, külön szögletes felület
  - Futásidőben dől el
- BoxInsetLayout
  - Négyzet alakú terület kör alakú órán is

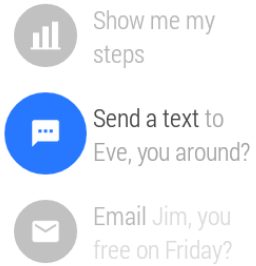


Felületi újdonságok

- Kártyák
  - Értesítések
  - Egyszeri interakciók
  - Csoportosított kártyák

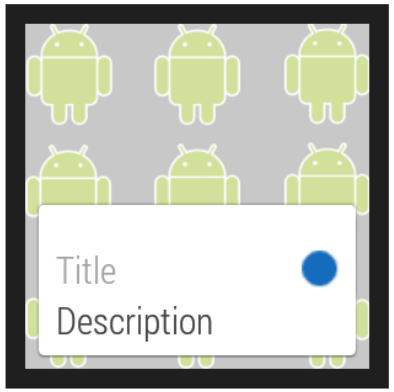


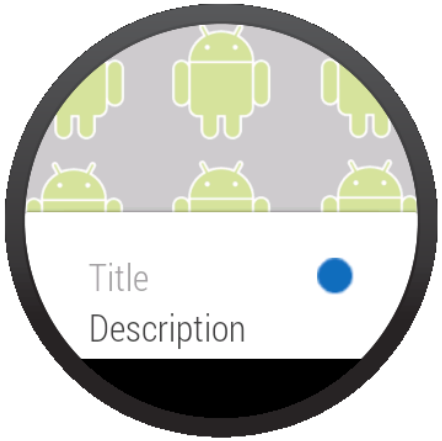
- WearableListView
  - Hasonló a már ismert ListView-hoz
  - Egyszerre egy aktív elem



Kártyák még

- ```
  <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">
    <FrameLayout
      android:id="@+id/frame_layout"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      app:layout_box="bottom" />
  </android.support.wearable.view.BoxInsetLayout>
  ```



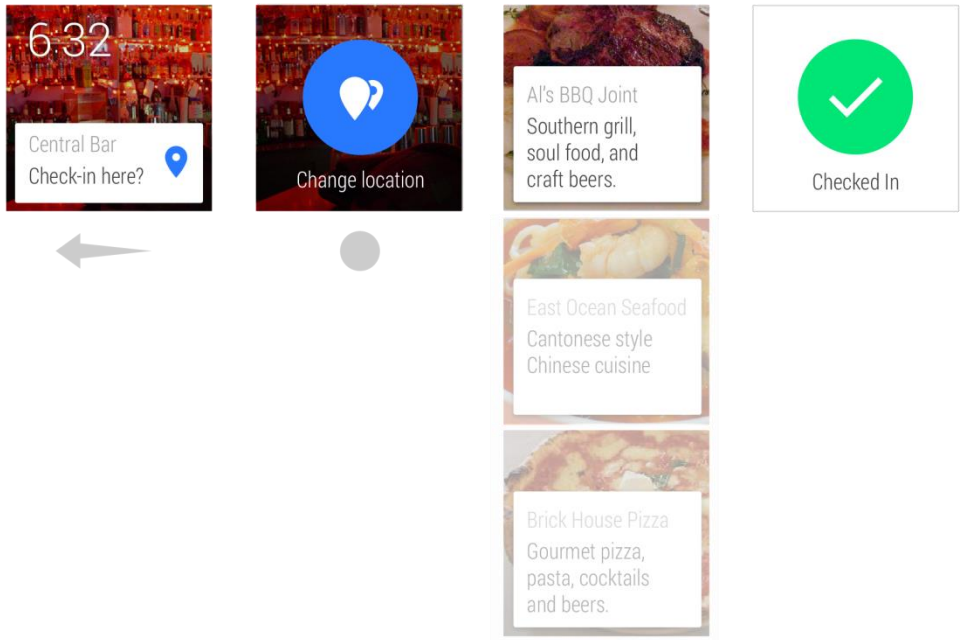



- ```
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_wear_activity);

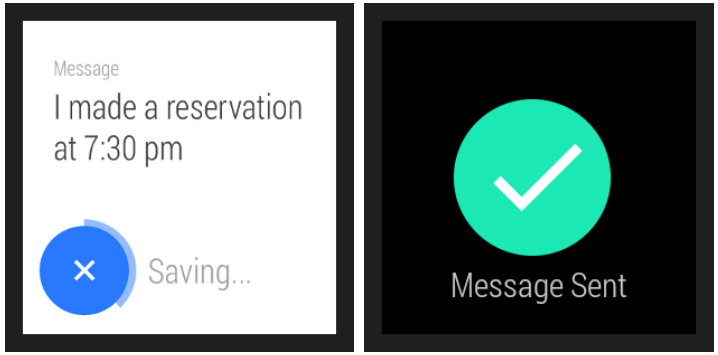
    FragmentManager fragmentManager = getFragmentManager();
    FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
    CardFragment cardFragment = CardFragment.create(getString(R.string.title), getString(R.string.desc),
  R.drawable.p);
    fragmentTransaction.add(R.id.frame_layout, cardFragment);
    fragmentTransaction.commit();
  }
  ```

Felületi újdonságok még

- 2D Picker
  - PageViewer + Kártyák



- Confirmations
  - Beépített időzítő
  - API által adott listener
  - ```
    Intent intent = new Intent(this, ConfirmationActivity.class);
    intent.putExtra(ConfirmationActivity.EXTRA_ANIMATION_TYPE,
                    ConfirmationActivity.SUCCESS_ANIMATION);
    intent.putExtra(ConfirmationActivity.EXTRA_MESSAGE,
                    getString(R.string.msg_sent));
    startActivity(intent);
    ```


    ```

Android Wear

Fejlesztői környezet beállítása

Wear emulátor

- Andorid 4.4W
- Teljes értékű Wear emulátor
- Valós eszköz és Wear emulátor közti együttműködés
- Szükséges port forward beállítás:

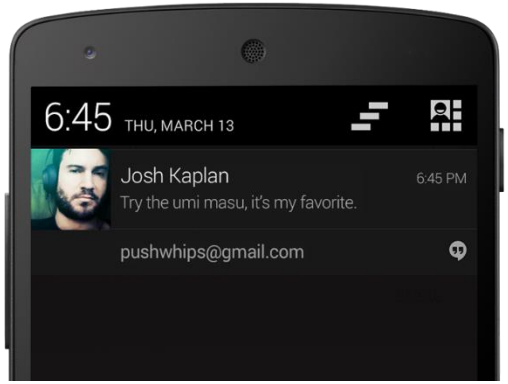
```
> adb -d forward tcp:5601 tcp:5601
```

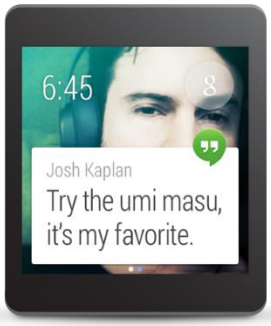
Android Wear

Értesítések

Értesítések

- Automatikus értesítés megosztás
  - > Nincs szükség külön Wear alkalmazás fejlesztésére!
- Értesítés -> kártya a „context stream”-en
- Wear specifikus akciók
- Hang alapú válaszlehetőség
- Több értesítés kezelése (notification stack)





Egyszerű értesítés

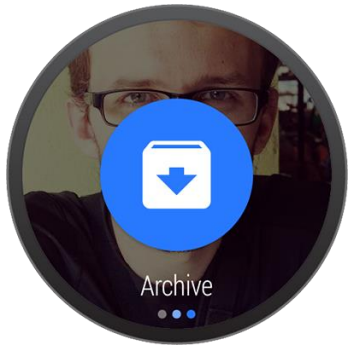
- Automatikus értesítés megosztás a Wear-el:
  `NotificationCompat.Builder`
- Gradle függőség:
  `compile "com.android.support:support-v4:20.0.+„`
- PendingIntent megadható:
  - Telefonon/tableten hajtja végre alapértelmezetten
- Notification megjelenítése:

```
NotificationManagerCompat notificationManager =
        NotificationManagerCompat.from(mContext);
// Issue the notification with notification
manager.
notificationManager.notify(notificationId,
notif);
```

További akciógombok

- *Builder addAction(...)* függvényével további akciók definiálhatók
  - > A készüléken egy újabb akciógomb az értesítéshez
  - > Wear-en egy új akció jobbra lapozáskor
- Lehetőség csak Wear akciók megadására:

```
WearableExtender.addAction()
```



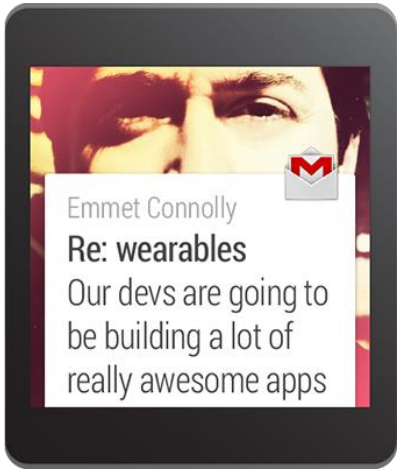
„Big view” értesítés

- Bővített tartalom az értesítéshez

  `NotificationCompat.BigTextStyle`

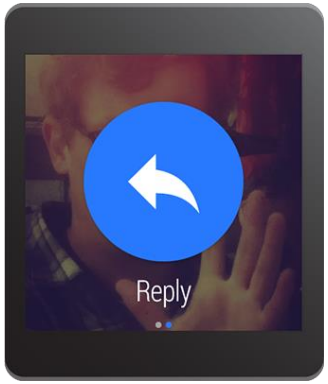
- Wear-en a BigView jelenik meg alapértelmezetten
- Háttér kép/ikon megadása:

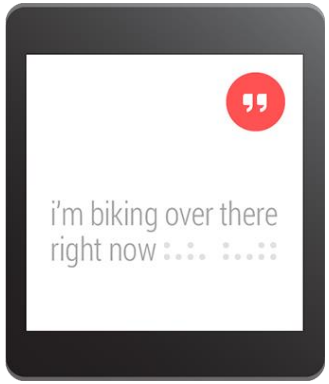
  `setLargeIcon()`

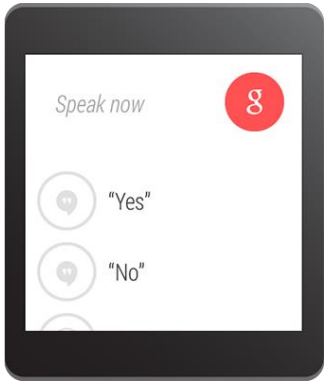


Válaszlehetőségek értesítésre

- Egyszerű válasz lehetőség
- Hang alapú válasz (billentyűzet nincs!)
- Előre definiált válaszok megadhatók
- A készülék oldalon a „válasz” intent-el kiolvasható az eredmény

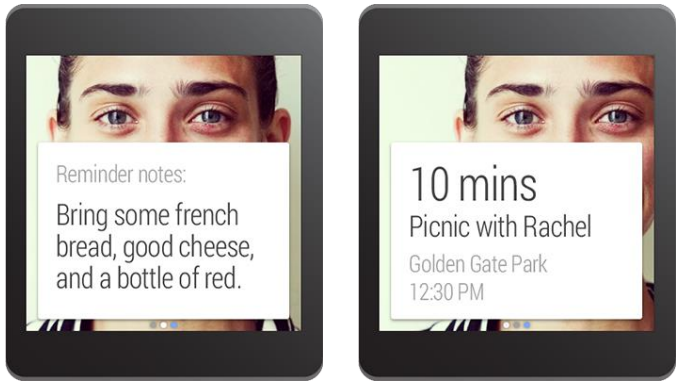






Több értesítés kezelése

- További információk megjelenítése esetén egy vagy több további oldal adható az értesítéshez
- Gyakorlatilag ez új lapok számára új *Notification* objektumot kell létrehozni
- *WearableExtender addPage(...)* vagy *addPages(...)* függvénye



Gyakoroljunk!

- Készítsünk egy alkalmazást, mely értesíti a Wear-t kimenő hívás esetén!
- Egészítsük ki a megoldást wear specifikus akciókkal!
- Adjunk hozzá hang alapú visszajelzési lehetőséget!

Android Wear

Kommunikáció

Kommunikáció

- Wearable Data Layer API
  - Google Play Services része
  - Bluetooth kommunikáció
  - Google ajánlás szerint nem használandó más
  - Minimum 4.3 (API 18+) a telefonon, tableten

```
GoogleApiClient mGoogleApiClient = new GoogleApiClient.Builder(this)
        .addConnectionCallbacks(new ConnectionCallbacks() {
                @Override
                public void onConnected(Bundle connectionHint) {
                    Log.d(TAG, "onConnected: " + connectionHint);
                    // Now you can use the Data Layer API
                }
                @Override
                public void onConnectionSuspended(int cause) {
                    Log.d(TAG, "onConnectionSuspended: " + cause);
                }
        })
        .addOnConnectionFailedListener(new OnConnectionFailedListener() {
                @Override
                public void onConnectionFailed(ConnectionResult result) {
                    Log.d(TAG, "onConnectionFailed: " + result);
                }
            })
        // Request access only to the Wearable API
        .addApi(Wearable.API)
        .build();
```

Wearable Data Layer

- DataItem
  - Egyszerű adatok tárolása
  - Automatikus szinkronizálás az eszközök között
  - DataApi-n keresztül használható
- Message
  - Egyszerű utasítások küldése (lejátszás, megállítás)
  - Nincs szinkronizálás, csak csatlakoztatott állapotban küldi el az üzenetet
  - MessageApi-n keresztül használható
- Asset
  - Bináris adatok, pl képek küldése
  - DataItem-hez csatolható
  - Automatikus cache és átvitel
- WearableListenerService
  - Broadcast figyeléséhez
- DataListener
  - Előtérben történő események figyeléséhez

Gyakoroljunk!

- Készítsünk egy alkalmazást, mely kérést küld a telefonnak!
- A telefon válaszban küldje el az elérhető szabad terület méretét!
- Az óra figyelje a választ és egy kártyán jelenítse meg a választ

Gyakoroljunk!

- Készítsünk egy telefon alkalmazást, mely szinkronizált adatokban tárolja az aktuális időjárást!
- Az óra olvassa ezeket az adatokat és ha van akkor jelenítse meg az adatokat.
- Az óra figyelje a választ és egy kártyán jelenítse meg a választ
- http://goo.gl/vo21gq

Gyakoroljunk!

- Készítsünk egy alkalmazást ami listázza a telepített alkalmazásokat!
- A kiválasztott elemhez tartozó alkalmazás induljon el

Forráskódok

- http://goo.gl/olwU18
- http://goo.gl/0df2I3
- http://goo.gl/HlehiC
- http://goo.gl/yQKCvm
- http://goo.gl/5JMOzB

Köszönöm a figyelmet!

nyitrai.istvan@bmeautsoft.hu