Mobile Workshop #5

Ex3 Remarks & Reminders

Code convention

Naming, Naming, Naming

- Packages: (this.is.a.package_name)
  - Bad : *com.examples.image_list_example*
  - Good : *il.ac.shenkar.remember_the_tahini*
- Classes: (ThisIsAClassName)
  - Bad: *TaskList* extends Activity
  - Good: *TaskListActivity* extends Activity
- Instances/Primitives:
  - thisIsAnInstanceName

Done Button – Not Recommended

```
<Button
    android:id="@+id/done"
    android:layout_width="match_parent"
    android:layout_height="fill_parent"
    android:onClick="done"/>
```

Done button - Wrong

```
public View getView(final int position, View convertView, ViewGroup parent) {
    final ViewHolder holder;

    if (convertView == null) {
        // ...
        holder.doneButton.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                TaskList.getInstance().removeTask(position);
                notifyDataSetChanged();
            }
        });

        convertView.setTag(holder);
    } else {
        holder = (ViewHolder) convertView.getTag();
    }
    // ...
}
```

Done button – Not great

```
public View getView(final int position, View convertView, ViewGroup parent) {
    ViewHolder holder;
    if (convertView == null) {
        // ...
    } else {
        // ...
    }
    // …

    holder.btdDone.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            itemDetailsrrayList.remove(position);
            notifyDataSetChanged();
        }
    });
    return convertView;
}
```

Done button – Good

```
public View getView(int position, View convertView, ViewGroup parent) {
    final ViewHolder holder;

    if (convertView == null) {
        // ...
        holder.doneButton.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                int pos = (Integer) view.getTag();
                TaskList.getInstance().removeTask(pos);
                notifyDataSetChanged();
            }
        });

        convertView.setTag(holder);
    } else {
        holder = (ViewHolder) convertView.getTag();
    }
    // ...
    holder.doneButton.setTag(position);
}
```

Done button – Holier than the pope

```
private final OnClickListener doneButtonOnClickListener = new OnClickListener(){
    @Override
    public void onClick(View view) {
        int position = (Integer) view.getTag();
        taskListModel.removeTask(getItem(position));
        notifyDataSetChanged();
    }
};

public View getView(int position, View convertView, ViewGroup parent) {
    final ViewHolder holder;

    if (convertView == null) {
        // ...
        holder.doneButton.setOnClickListener(doneButtonOnClickListener);
        convertView.setTag(holder);
    } else {
        holder = (ViewHolder) convertView.getTag();
    }
    // ...
    holder.doneButton.setTag(position);
}
```

Broadcast Receivers

- Allows message sending (broadcasts) from Android to your app.
- Broadcasts can be received even if the app is closed.
- Register to system events.
- Register to application events.

Broadcast Receiver

```
public class ReminderBroadCastReceiver extends
BroadcastReceiver {
  public void onReceive(Context context, Intent   intent) {
    //do something QUICK
  }
}
```

- BroadcastReceiver object is only valid for the duration of the call to onReceive
- anything that requires asynchronous operation is not available (You can by using goAsync())

BroadcastReceiver: statically AndroidManifest.xml

```
<receiver android:name=".ReminderBroadCastReceiver">
  <intent-filter>
    <action android:name="com.rtt.reminder_broadcast" />
  </intent-filter>
</receiver>
```

BroadcastReceiver: Dynamically register

- Context.registerReceiver()
- Context.unRegisterReceiver()
- If registering a receiver in your Activity.onResume() implementation, you should unregister it in Activity.onPause()
- **Don’t forget to unregister your receiver -prevent Memory leak**

BroadcastReceiver Security

- The Intent namespace is global (Conflict problem)
- any application may send broadcasts to that registered receiver.
- any other application can send broadcasts to it regardless of the filters you specify. To prevent others from sending to it, make it unavailable to them with android:exported=“false".
- You can control who can receive such broadcasts through permissions
- Read more here

LocalBroadcastManager

- Helper to register and send broadcasts of Intents to local objects within your process.
- the data you are broadcasting won't leave your app
- It is not possible for other applications to send these broadcasts to your app
- It is more efficient than sending a global broadcast through the system.

Intent & PendingIntent

- Intent: I want to do something.
- PendingIntent: I want it to happen later.

Alarm Manager

- Access the android system alarm services
- Set alarms for when your app is not open
- Not to be confused with timers within your application. (see **Handler** for that)

Set an Alarm for your BroadcastReceiver

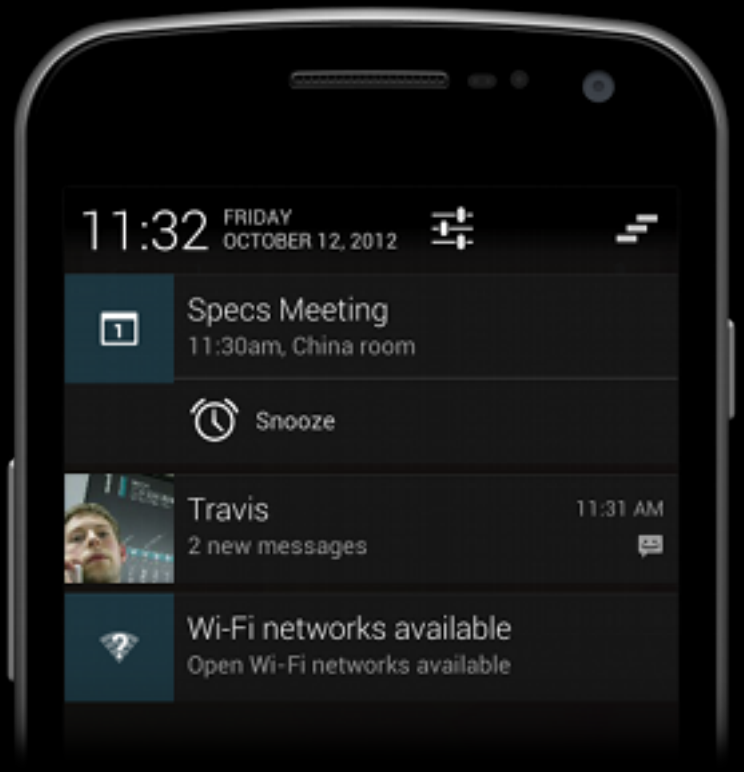
```
Intent intent = new
Intent("com.rtt.reminder_broadcast");

PendingIntent pendingIntent =
PendingIntent.getBroadcast(context, 0, intent, 0);

AlarmManager alarmManager =
(AlarmManager)getSystemService(ALARM_SERVIC);

alarmManager.set(AlarmManager.RTC_WAKEUP,
System.currentTimeMillis() + 20000, pendingIntent);
```

NotificationManager



Show a notification

Intent myIntent = new Intent(context, TaskListActivity.class);
PendingIntent pendingIntent =
PendingIntent.getActivity(context, 0, myIntent, 0);

NotificationManager notificationManager =
(NotificationManager)context.getSystemService(Context.NOTIFICATION_SERVICE);

```
// Build notification
        Notification noti = new Notification.Builder(context)
                  .setContentTitle(message).setContentText(message)
                  .setSmallIcon(R.drawable.ic_launcher).setContentIntent(pIntent)
                  .build();
        notifiManager.notify(0, noti);
```

HINTS

- You can pass messages in your Intents with putExtra() and getExtra()

Now you

- BroadcastReceivers (the bottom example is the important one): http://www.vogella.com/articles/AndroidBroadcastReceiver/article.html
- NotificationManager (simple) :http://android-er.blogspot.co.il/2011/04/simple-example-to-send-notification.html
- NotificationManager (advanced): http://www.vogella.com/articles/AndroidNotifications/article.html
- Ex5