

Prof. Dr. Michael Rohs, Dipl.-Inform. Sven Kratz

## Mensch-Maschine-Interaktion 2 (Mobile Interaktion) WS 2011/2012

### Übungsblatt 7

#### Aufgabe 1: Android-Client für SoundSquare [Einzelabgabe]

In diesem Übungsblatt geht es um die (teilweise) Implementierung eines Android-Clients für SoundSquare, für denen Sie im letzten Übungsblatt einen Paper-Prototypen entwickelt haben.

#### Beschreibung der Funktionalität

Der SoundSquare-Server, der bereits existiert, speichert die Orts- und Audiodaten. Details dazu finden sich in den Folien Übungsblatt auf der Vorlesungs-Webseite. Ihre Anwendung soll folgende Funktionalität bieten und die folgenden Nebenbedingungen erfüllen:

1. Die Anwendung zeigt eine geographische Karte an, auf welcher der aktuelle Aufenthaltsort des Benutzers („You are Here“-Punkt) sowie Orte mit Audiodaten („Points of Interest“, POIs) angezeigt werden. Ein Beispiel zeigt die untenstehende Abbildung.
2. Der Benutzer kann die Karte bewegen und in die Karte hinein- bzw. herauszoomen.
3. Der Benutzer kann einstellen, dass die Karte dem „You are Here“-Punkt folgt.
4. Klickt der Benutzer auf den „You are Here“-Punkt, dann wird an dieser Stelle ein neuer POI angelegt. Ein POI hat eine Position, ein Erzeugungsdatum, einen Namen und ein Tag. Ein Tag ist ein Schlüsselwort für die Kategorie des POIs, z.B. „Kreuzung“.
5. Klickt der Benutzer auf ein POI, dann werden die Metadaten dieses POIs angezeigt, sowie die Liste der Sounds, die für diesen POI gespeichert sind. Außerdem gibt es die Möglichkeit, für diesen POI einen weiteren Sound aufzunehmen.
6. Für einen Sound kann der Benutzer vor der Aufnahme eine Beschreibung eingeben. Ein Sound hat außerdem ein Erzeugungsdatum.
7. Erzeugte POIs und Sounds werden automatisch auf dem SoundSquare-Server gespeichert. Für einen geographischen Bereich lässt sich die Liste der POIs vom Server erfragen. Für einen POI lässt sich die Liste der Sounds vom Server erfragen. Neu erstellte POIs und Sounds werden automatisch zum Server hochgeladen. Der Server existiert bereits. Details zum Kommunikationsprotokoll mit dem Server sind auf der Vorlesungs-Webseite in den Folien zur Aufgabe zu finden.
8. Bei einem Lokations-Update erfragt die Anwendung jeweils die aktuelle Liste der POIs für diesen Bereich vom Server.

#### Strukturierung der Android Applikation

Ihre abzugebende Android Applikation sollte mindestens aus den folgenden Teilen bestehen. Falls Sie die oben genannte Funktionsbeschreibung in ihrer Paper-Prototyping-Phase anders umgesetzt haben, darf sich die Struktur Ihrer Applikation auch dementsprechend unterscheiden:

- eine Hauptactivity, die die Karte anzeigt, sowie die POIs und den „You are here“-Punkt als Overlays darstellt. Die Hauptactivity sollte die Funktionalität von Punkten 1-5 bieten. (*Hinweis:* erweitern Sie dazu die Klasse `MapActivity`.)
- eine Activity, in der Sie einen Audiorekorder implementieren. Sie können hierzu die Implementierung der Aufnahmefunktionen `recordaudio()` und `stopRecordingAudio()` in `SoundRecorderActivity.java` als Grundlage nehmen.
- Zum abspielen der Sounds benötigen Sie eine Activity, die eine Liste von Sounds zu einem gegebenen POI anzeigt, und einzelne Sounds bei Berührung des Listeneintrages abspielt. Sie können zur Implementierung dieser Activity die teilweise implementierte Klasse `SoundPlayerActivity.java` verwenden.
- Letztlich braucht die App noch eine Activity, in der neue POIs angelegt werden können.

### Weitere Hinweise zur Bearbeitung:

- **Vergessen Sie nicht die Materialien zu ihrem Papier-Prototypen aus Übung 6 mit abzugeben.**
- Sie können die von uns bereitgestellte `AndroidManifest.xml` als Grundlage für diese Übung verwenden.
- Machen Sie sich zur Implementierung der Kartenansicht und der Overlays mit dem Google Map View vertraut:  
<http://developer.android.com/resources/tutorials/views/hello-mapview.html>
- Erstellen Sie einen eigenen Google-Map-Key, wie in den Vorlesungsfolien beschrieben.
- Die Klasse `Server.java` implementiert die notwendige Funktionalität zur Kommunikation mit dem Server. Verwenden Sie diese Klasse zur Kommunikation mit dem Server.
- Vergessen Sie nicht, die notwendigen Permissions und Bibliotheken im `AndroidManifest` anzupassen, falls notwendig.
- Die Aufgabe kann komplett mit dem Emulator gelöst werden. Dem Emulator können Längen- und Breitengrad, wie in den Vorlesungsfolien beschrieben, mitgeteilt werden.
- Im Emulator müssen „GPS support“, „Audio playback support“ sowie „Audio recording support“ aktiviert sein. Außerdem muss der Emulator eine (virtuelle) SD-Karte besitzen. Die Hardware-Optionen lassen sich im Android Device Manager ändern. Evtl. hilft es, ein bestehendes device image zu löschen und mit den richtigen Optionen neu anzulegen.
- Um „mock locations“ auf einem Android-Mobiltelefon zu benutzen muss „Falsche Standorte“ in Einstellungen | Anwendungen | Entwicklung aktiviert sein. Einige Apps im Android Market erlauben es, „mock locations“ zu erzeugen (Suche nach „mock location“).
- Die Klasse `SoundInfo.java` speichert jeweils die Daten zu den einzelnen Sounddateien, die einem POI zugeordnet sind. `PointOfInterest.java` dient zur Speicherung von Daten zu den POIs. `PointOfInterest` und `SoundInfo` Instanzen implementieren `Serializable`, und können somit zwischen Activities ausgetauscht werden.
- Sie werden möglicherweise Daten zwischen verschiedenen Activities austauschen müssen (z.B. selektierter POI von der Hauptaktivität an `SoundPlayer`). Machen Sie sich mit den in Android vorhandenen Mechanismen vertraut, die den Austausch von Daten zwischen Activities unterstützt:
  - o <http://thedevelopersinfo.wordpress.com/2009/10/15/passing-data-between-activities-in-android/>
  - o <http://developer.android.com/reference/android/app/Activity.html> („Starting Activities and Getting Results“)

- Insbesondere können beim Aufruf einer Activity serialisierte „Extras“ mit der Methode `getSerializableExtra("foo")` wieder hergestellt werden:

```
protected void onActivityResult(    int requestCode,
                                   int resultCode,
                                   Intent data ) {
super.onActivityResult(requestCode, resultCode, data);
...
data.SoundInfo si = (SoundInfo) data.getSerializableExtra("soundInfo");
... }
```

## **Abgabe**

**Achtung:** Plagiate sind verboten und führen zum Ausschluss aus der Veranstaltung!  
Das Programm muss kompilieren, sonst wird es nicht korrigiert! Dieses Übungsblatt muss einzeln bearbeitet werden. Geben Sie die Übung als zip-Datei (Eclipse, Export → Archive file) bis Montag, den 19.12.2011 um 12:00 Uhr im UniWorX Portal (<https://uniworx.ifi.lmu.de/>) ab. Sie sollten Ihre Lösung in der Übung vorstellen können.