

Script generated by TTT

Title: Klgel: Assignment2 (21.04.2014)

Date: Mon Apr 21 02:46:33 CEST 2014

Duration: 23:52 min

Pages: 16


Applied Informatics /
Cooperative Systems
AICOS
Prof. Dr. J. Schlichter

Technische Universität München

TUM

Introduction Assignment 2

Niklas Klügel



Introduction

After Assignment 1 you should have elaborated a concept for Location-Based Social Game:


- Use some individual context of the user and his/her teammates (e.g. use localization data)
- Use the long-term Social Context of the user (e.g. Social Networks)
- Use of a short-term Social Context of the user (e.g. interaction with other users, localization)

Outline of this talk:

1. Presentation Part:

- Assignment 2
- Introduction to provided exemplary prototypical implementation

2. Practical Part:

- Set up of development environment
- 


Assignment 2

Assignment 2 is to implement your concept of a Location-Based Social Game into a working prototype:

- As an Android Application (formal requirement)
- Potentially using a central Server for storing/processing user and game data

Note: you are free to use any online resources, as long as they are documented (cite)!

To get you started, we have created an exemplary implementation of a (simple) game:

- Makes use of long & short term Social Context, individual Context
 - Utilizes different modes of communication (for events, user messages, ...)
 - Provides infrastructure to handle user & localization data and to process them further
 - Is divided into Client & Server Application(s)
- 

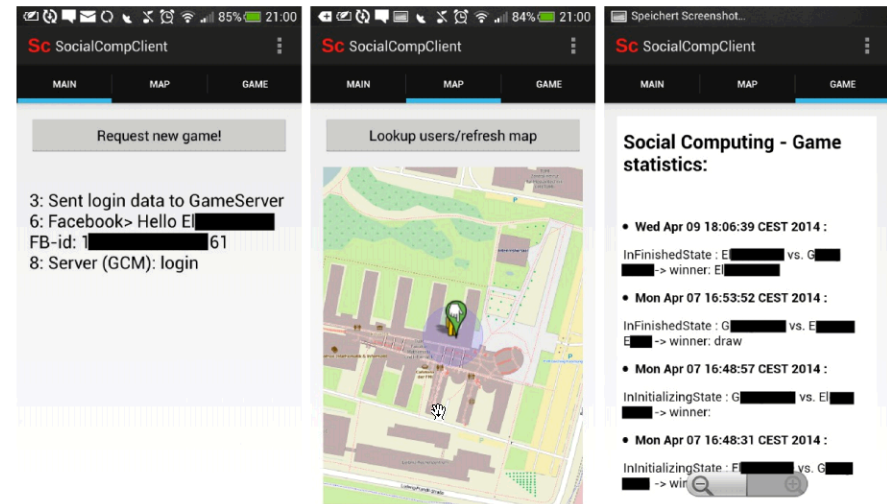
What is provided (game play)?

After geo-locations and the long-term Social Context (Social Network) of the users have been acquired:

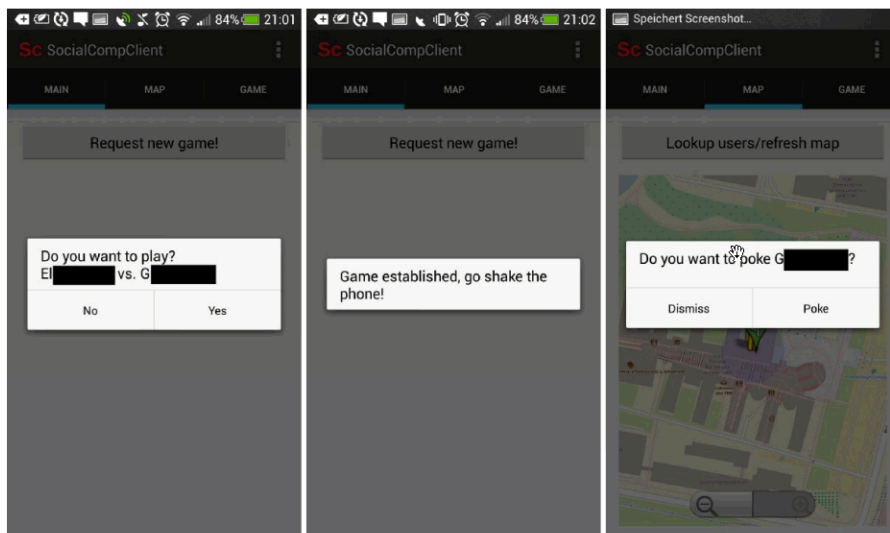
- A user can request a new game,
- A fitting opponent is chosen, conforming the following criteria (to simplify testing):
 - is a friend of the requesting user
 - Is near requesting user within 100 meters
- If both users accept the game, they are asked to shake their phones
(= confirmation of short-term Social Context)
- The winner is randomly chosen and then given some credits on his/her score
- If the game is not accepted by a user, some points will be subtracted from his/her score

Note: This serves as a rough outline for a game play, you are free to deviate from it!

Screens in the application (client)



Dialogs in the application (client)



Application Design

Client-Server Application:

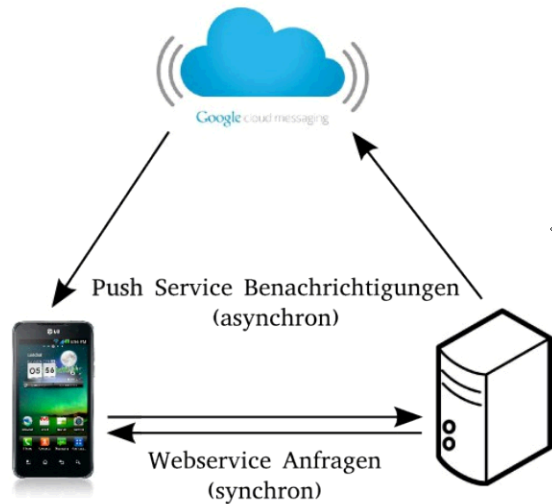
- Server: Webservice that contains all game logic and handling of necessary data
 - To circumvent malicious intend (malicious clients)
 - To respect privacy concerns (not exchanging private data between clients)
- Client: mobile Application that predominantly serves as a User Interface to the services provided by the server (ThinClient!)
 - Triggers events
 - Reacts to external events

Client can contact Webservice directly, but how does the Webserver contact Clients?

- Client IP Addresses change
- Or are shared (NAT)

→ Use Push Services (provided by Google)

Communication channels



What is provided (functional design)?

Webservice:

- Login/Register Users
- Updating geo-locations in a database
- Looking up nearby users to a geo-location
- Initializing a game while respecting criteria of location and social context
- Sending/accepting/aborting a game (request)
- Evaluating the social interaction
- Administering the users' data
- Administering device information (for push messaging)
- Sending messages via Push messaging

Client:

- Sending game requests
- Showing dialogs and sending conditionally messages to the server
- Reading accelerator sensor data (for shaking the phone)
- Receiving/processing Push messages
- Show positions of other users in a map
- Unidirectional messages ("poke" another user)

Involved Technologies

Main motivators:

- Get you started quickly (simple installation/setup/API)
- Good documentation/resources

Used Technologies:

- Social Network (Facebook)
- Android 4.2+ (Client)
- Play Framework 2.2 (Webservice)
- MongoDB (Database)
- JSON (Client-Server communication, data format)
- HTTP (Client-Server communication, transport protocol)
- Map Data (OpenStreetMaps)

RESTful Client-Server communication:

- Client does not save game/user data
- Webservice is not concerned with user interaction (UI)
- Webservice is stateless w.r.t. clients

Involved Technologies

Facebook:

- Most widely used, very good documentation for projects and Apps
- Client: Official Facebook SDK for Android
- Server: restFB (since no official SDK exists), widely adopted

Note: You are not forced to use Facebook, use whatever fits

- Diaspora?
- Roll you own simple Social Network on the Webservice

Involved Technologies

Play Framework:

- Complete development stack, including Webserver, WebApp, Database, Dependency Management (sbt), Asynchronous I/O,...
- Very little configuration necessary to get going
- Uses Hot Code Reloading, helps speed up development
- Has support for most IDEs
- Straightforward API, superb documentation

Involved Technologies

Google Play Service with Push Messaging:

Bound to use this for the Android Platform, still good documentation

OpenStreetMaps:

- Less restrictive in usage than Google Maps (e.g. number of requests)
- Similar quality for European Map Data
- Basic functionality of Map Visualization similar to Google Maps

Communication

- Synchronous communication (initiated by clients) using HTTP+JSON
Functions served are accessible via simple URLs with parameters
- Asynchronous communication (initiated by the server) using Push Messaging+JSON

Most complex process is the login/registering of a new user as it involves several parties.

Further Outline

Practical part of this talk is intended to help you set everything up:

1. Get the source code for the projects
2. Set up MongoDB
3. Install Play
(quick test of webservice)
4. Install Android development environment
5. Set up the client project and SDKs
6. Create/register a Facebook App
7. Create/register a Google App
8. Quick overview over the source code

