



Distributed Systems 2015 – Open Project

Anwar Hithnawi
hithnawi@inf.ethz.ch



Open **P**roject

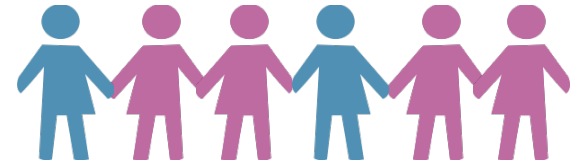
Project

- Find a partner group
Team up with up to **6 students**
- Choose your own topic
 - **Constraints:** Distributed component & Android
 - E.g., Architecture, Synchronization, Concurrency, Consistency, Distributed commit, Consensus.
- Submission
 - Project proposal
 - Project code
 - Slides for the 1-minute madness



Register your team

- Form groups of up to 6 students each
 - Not less than 3
- Via the submission system
 - Create a new group
 - Add members
 - Submit project deliverables (code and project proposal)

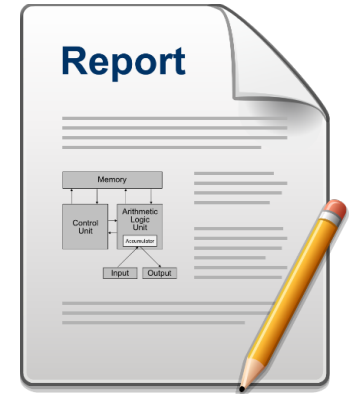


Submission deadline



- Project proposal
 - Deadline: **13th November 2015, 9:00 AM**
 - Submission system
 - Naming convention: “*report.pdf*”
- Presentation slides
 - Deadline: **17th December 2015, 14:00**
 - By e-mail: hithnawi@inf.ethz.ch
 - E-mail subject: “[2015] Distributed systems - <group_leader_nethz>”
 - Format: only PDF (naming: “*vs-nethz-presentation.pdf*”)
- Code
 - Deadline: **18th December 2015, 9:00 AM**
 - Submission system
 - Naming convention: “*code.zip*”

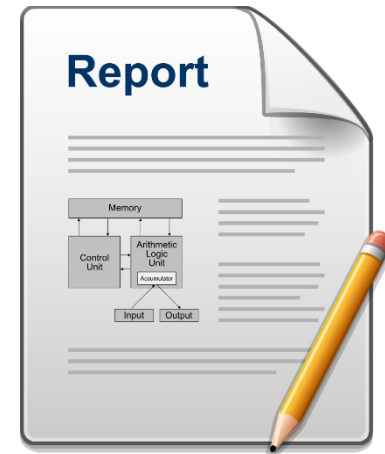
Project Proposal



- Conclude the initial planning phase
 - This is where you choose a project, set your goals, clarify your ideas, and find the material you will need.
- An opportunity to take an early feedback on your project idea

Project Proposal

- Only one report per project (3 – 4 pages)
- Focus on technical description of your work
 - Problem statement
 - System overview (e.g., architecture)
 - Clearly state the distributed systems components
 - Requirements
 - E.g., external libraries, hardware
 - Work packages
 - Planning: Schedule + distribution among team members
- Only LaTeX allowed! (template provided)



Report tips

- The report should have a technical report style
 - Formal language style
 - Try to avoid: “Over the course of the semester we learned a great deal, both directly from our professors ...”
- Abstract
 - Problem statement, project overview, expected deliverables
 - Focus on the novel elements
- Use meaningful section names
- Avoid code in the report, unless very important
 - Use pseudocode (easier to follow and read)
- Any figures/tables must be referenced from text

Abstract example

- Bad abstract

ABSTRACT

Throughout this project a Samsung Galaxy S2 with API 16 was used.

ABSTRACT

We used the Samsung Galaxy Nexus (running Android 4.3), the Asus Nexus 7 (running Android 4.4) as well as the Sony Xperia Tipo Dual (running Android 4.0) to build a 2D artillery game.

- Good abstract

ABSTRACT

We present a cross-platform game called Tronium that allows up to eight players to play together via local network, or alternatively allows single-player matches against AI opponents. Tronium is inspired by the "light cycle" scene from the 1982 film "Tron" and is implemented using the Unity[®] engine, which is a high-level framework for game development. The game supports Windows, Mac OS and Linux on x86/x86_64 and Android[™] with potential for easy ports to others platforms thanks to the cross platform capabilities of the Unity engine.

Project Presentation

- Prepare slides for 1-minute madness
- Focus on selling your idea
 - Make clear what your app does, why someone would need it and what is nice about it
 - Motivation, general idea, interesting technical aspects, results, ...
- Include a live demo whenever suitable
- 1-minute madness will take place on 18th December, 2015



Demo session

- Demo session will follow the 1-minute madness
- Similar to an exhibition booth
 - Possibility to discuss with others and answer questions
- Inform us early enough about any special requirements for your demo
 - E.g., need Internet for your demo?
- All students are required to participate to be graded
 - Conflicts: Make sure at least one member can attend and demonstrate the project

Selected projects from previous years

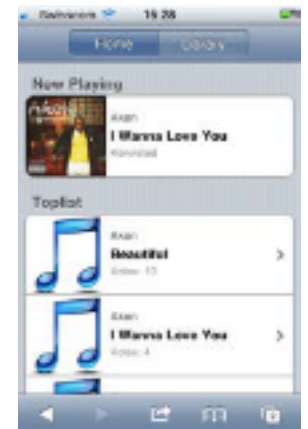
Examples



djCrowd – Interactive distributed music player

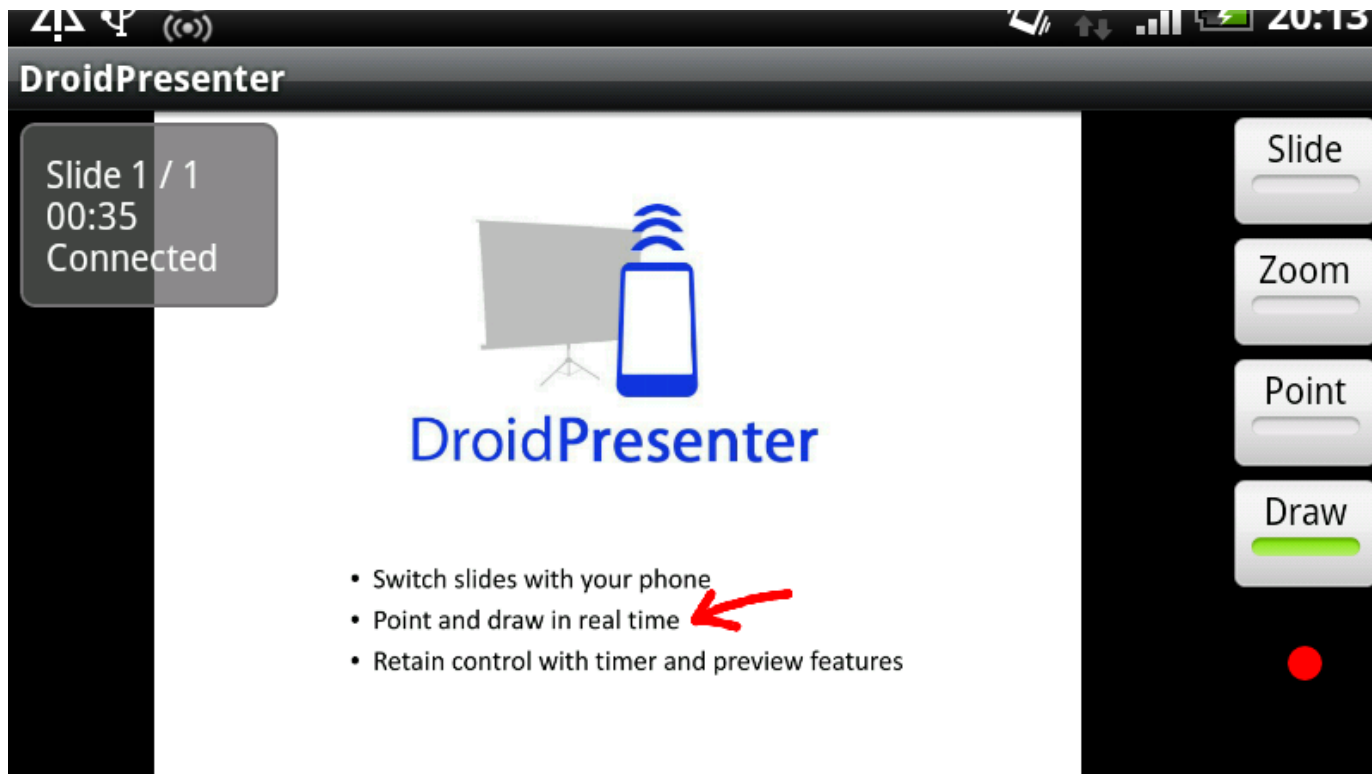
HS10: Luchin Doblies, Alexander Grest, Moritz Hoffmann, Jost Joller, Philipp Schmid, David Stolz

- Start up one phone as server
(connected to hi-fi system)
 - Your friends can connect to the server
 - Check the song that is currently playing
 - See upcoming songs in the playlist
 - Modify playlist by voting for their preferences
 - Upload songs from their phones
- + Web interface to provide access for non-Android devices



DroidPresenter – Presentations remote control

HS10: Andreas Tschofen, Leonhard Helming, Mathias Buerki, Damian Karrer



DroidPresenter allows you to draw in, point at, zoom in/out and control your presentation through your smartphone

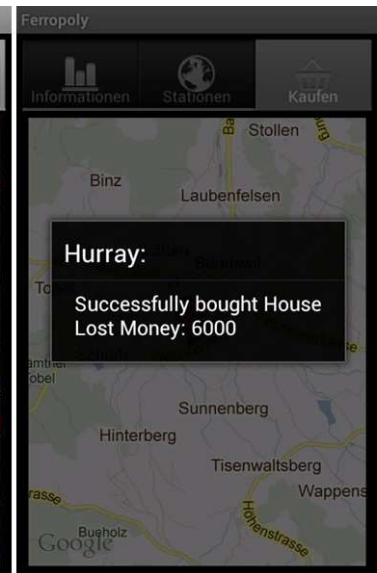
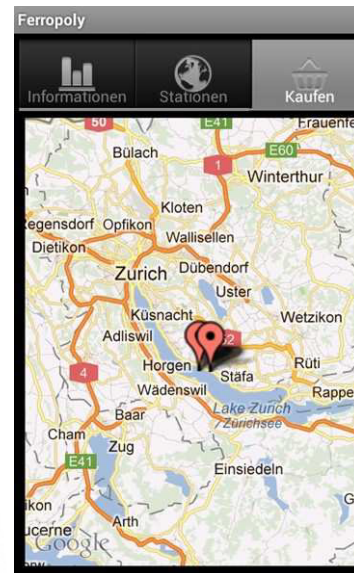
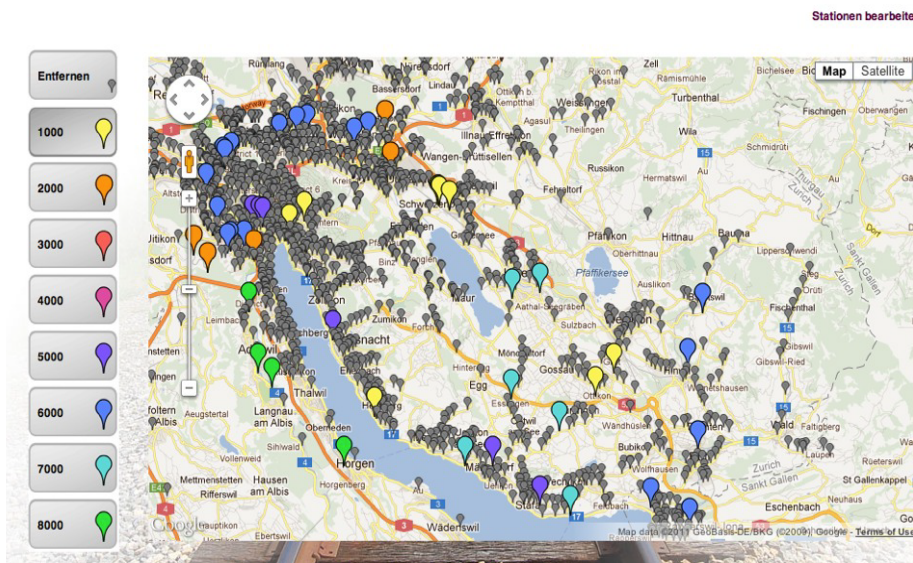
Ferropoly – Monopoly in the real field

HS11: Ameri Michael, Aras Ersan, Marti, Messmer Stefan

FERROPOLY



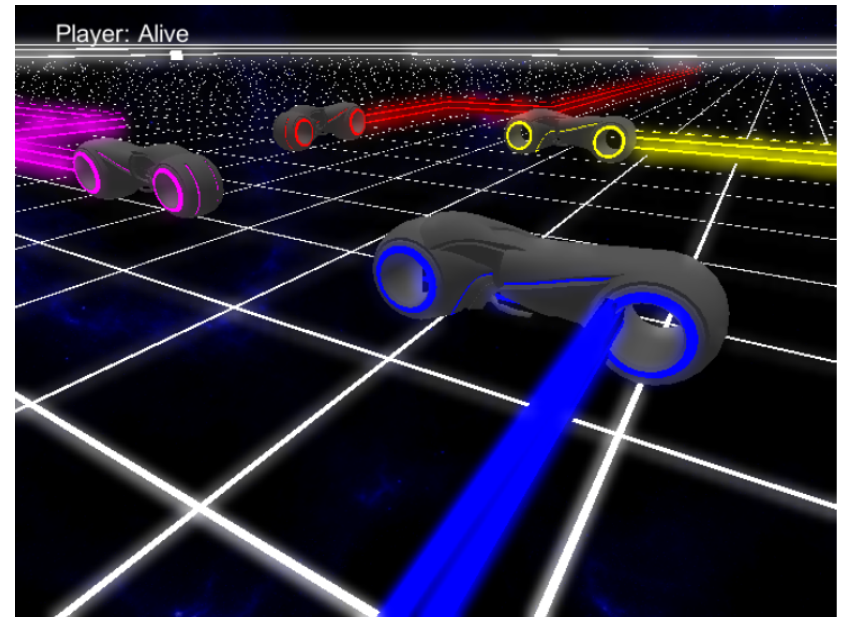
- Emulate Monopoly in the real world
 - Travel across Switzerland and buy train stations
 - Ruby on Rails server
 - REST services with JSON interface



Tronium – Cross platform game

HS13: Lukas Häfliger, Alexandra Maximova, Thomas Müller, Christian Vonrüti, Alexander Viand, Marko Živkovic

- Based on the Tron movie
- Up to 8 players
- Over local network
- AI players
- [Unity Game Engine](#)



Jass card game

HS13: Fabian Stutz, Jannick Griner, Priska Pietra, Dejan Mircic, Michael Franz, Nicolas Forster

- Client-Server architecture
- Server = tablet
- Clients = mobile phones

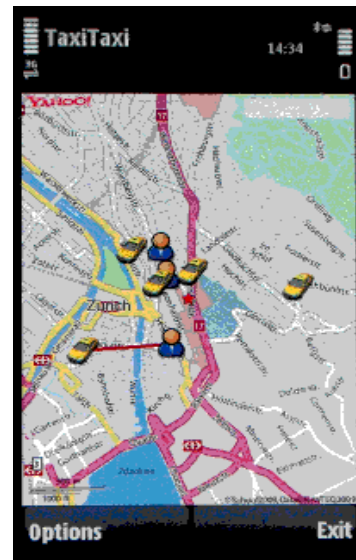
- 3 main components to consider:
 - Networking
 - Game logic
 - GUI

- Similar approach for other card games



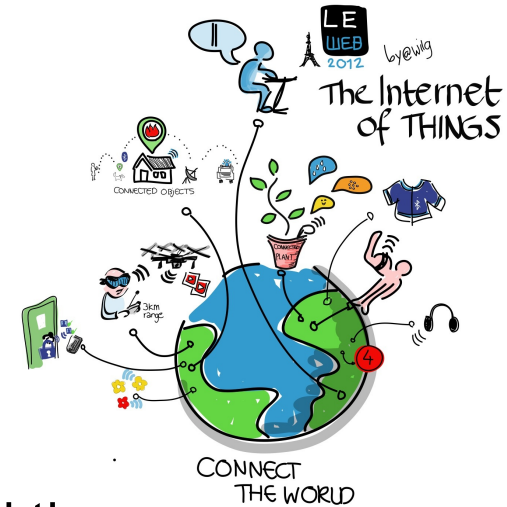
Consensus-based Taxi

- Implementation for the consensus problem
- Distributed application to find the optimal cab



Internet of Things Development Platforms

- We will provide IoT hardware for 4 projects
 - Estimote Beacons
 - TI SensorTag
- Make a dummy object smart
 - Change how people interact with the world around them
- “The Internet of Things (IoT) is the network of physical objects embedded with electronics, software, sensors, and network connectivity.”
- Create opportunities for more direct interaction between the physical world and virtual world



Estimote Beacons

- ARM processor, BLE radio, sensors (accelerometer & temperature)
- iBeacon and Eddystone compatible
 - BLE device periodically broadcasting a unique ID
 - Location awareness applications



- Useful Links:

- <http://developer.estimote.com/>
- <https://github.com/Estimote>
- Google Beacons: <https://developers.google.com/beacons/?hl=en>
- <https://github.com/google/eddystone>
- <https://www.youtube.com/watch?v=SrsHBjzt2E8>



TI SensorTag

- ARM processor, BLE radio, 10 sensors
 - Light, digital microphone, magnetic sensor, humidity, pressure, accelerometer, gyroscope, magnetometer, object temperature, and ambient temperature

- Links:

- <http://www.ti.com/tool/cc2650stk>
- <https://store.ti.com/cc2650stk.aspx>
- http://processors.wiki.ti.com/index.php/CC2650_SensorTag_User%27s_Guide





Final remarks

- We recommend you to use Control Version Systems (e.g., Git, Mercurial or SVN)
 - Github: <https://github.com/>
 - Slides for the Git-tutorial <https://docs.google.com/presentation/d/1BbLSI-ef7dMi2m1JkWTn0fqjbXGo-il8sFQVr9LtUUc/edit#slide=id.p>
 - Introduction to Git: <http://git-scm.com/book>
 - VIS GitLab: https://gitlab.vis.ethz.ch/users/sign_in
- Deliverables
 - Project Proposal (3 – 4 pages, “*report.pdf*”)
 - Code (naming convention: “*code.zip*”)
 - Slides for 1-minute madness (“*vs-nethz-presentation.pdf*”)
- Grading
 - Project proposal, presentation, implementation, complexity, innovation

Have Fun Programming!

