Multiplayer Pacman

Distributed Systems – Project Proposal

Student One, Student Two, Student Three
ETH ID-1 XX-XXX-XXX, ETH ID-2 XX-XXX-XXX, ETH ID-3 XX-XXX-XXX
one@student.ethz.ch, two@student.ethz.ch, three@student.ethz.ch

ABSTRACT

(a) System overview *(b)* software and hardware you intend to use in this project, *(c)* expected deliveries of this project.

1. INTRODUCTION

Since 1980, the game Pac-Man has fascinated players around the world. Starting as an arcade game, it was adapted for many platforms while technology was improving and is still played today on mobile devices. While graphics have improved significantly over time and new features were added to the original idea of the game, there is still a major drawback to most of its versions as the game only provides single player user experience.

Our goal is therefore to exploit the opportunities of modern mobile devices in order to create a new user experience. This new user experience will consist of the well known Pac-Man game combined with a distributed multi player approach. The idea will be implemented as an android application allowing the user to create a new instance of the game by starting a server on an android device. Other users can then connect to this server with their own devices.

2. SYSTEM OVERVIEW

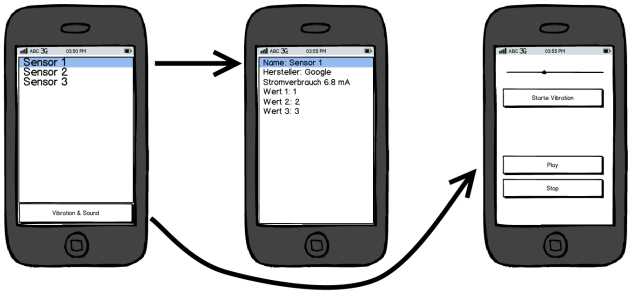

Figure 1: Only include useful figures. Do not simply copy something from a Web.

3. REQUIREMENTS

1. The game can be played on multiple Android devices (at least two).
2. The gameplay should work as follows:
   (a) Coins are distributed evenly on the game map (board).
   (b) One player plays as PacMan
   (c) One or multiple other players play as ghosts
   (d) PacMan wins, if he collects all coins on the map
   (e) The ghosts win, if they capture PacMan (simply modeled by collision).
3. Each player must use one Android device in order to control his figure (PacMan or ghost)
4. The map (board) on which the players move should provide the following features:
   (a) PacMan starts on a predefined location (PacMan spawn)
   (b) The ghosts (one ore multiple) start on predefined locations (ghost spawns)
   (c) Player figures can only move up, down, left and right.
   (d) The only structuring elements of the map are walls.
   (e) Walls have eiter horizontal or vertical orientation.
   (f) Player figures can not move through walls.
   (g) The map has rectangular shape and is limited by walls at its borders (horizontal walls along the left and right border, vertical walls along upper and bottom border).
   (h) Every position at the map that is not occupied by a wall must be reachable for the player figures.
5. The protocol used for Device-to-Device communication will will be implemented atop UDP, TCP or a higher-level protocol [???]
6. The app will be optimized for Android version [???] and run on Android version [???] and higher. [???]
7. The player who hosts the game will also play as Pac-Man.
8. On the hosting player's device, a server task will be started that clients can connect to.
9. The players that connect to a hosted game play as ghosts.

The Project

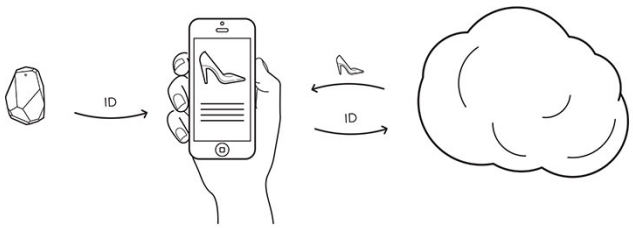

Figure 2: System Overview [1]

4. WORK PACKAGES

- **WP1**: Define appropriate model for game situation (state) and methods to change the state.
- **WP2**: Implement a map generator.
- **WP3**: Design graphical representation for PacMan player figure.
- **WP4**: Design graphical representation for Ghost player figure.
- **WP5**: Design graphical representation for map elements (coins, walls, unoccupied positions).
- **WP6**: Implement Game activity that connects model and graphical representation. It should also allow the user to control it's figure.
- **WP7**: Define a communication protocol to synchronize the game's state accross devices. Choose appropriate transport protocol.
- **WP8**: Implement the communication protocol - part 1: Server.
- **WP9**: Implement the communication protocol - part 2: Client.
- **WP10**: Design and implement start menu that allows the players to connect to the game.
- **WP11**: Combine Game activity with the communication protocol.

Stick to a concise, scientific writing style.

5. MILESTONES

6. REFERENCES

[1] Estimote. `http://estimote.com/`. Accessed on 26 Oct 2015.

[2] Services: Sending Notifications to the User. `http://developer.android.com/guide/components/services.html#Notifications`. Accessed on 29 Aug 2013.

[3] E. Burnette. *Hello, Android: introducing Google's mobile development platform*. Pragmatic Bookshelf, 3 edition, 2010.

[4] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, 1999.

[5] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. Phd thesis, UC Irvine, 2000.