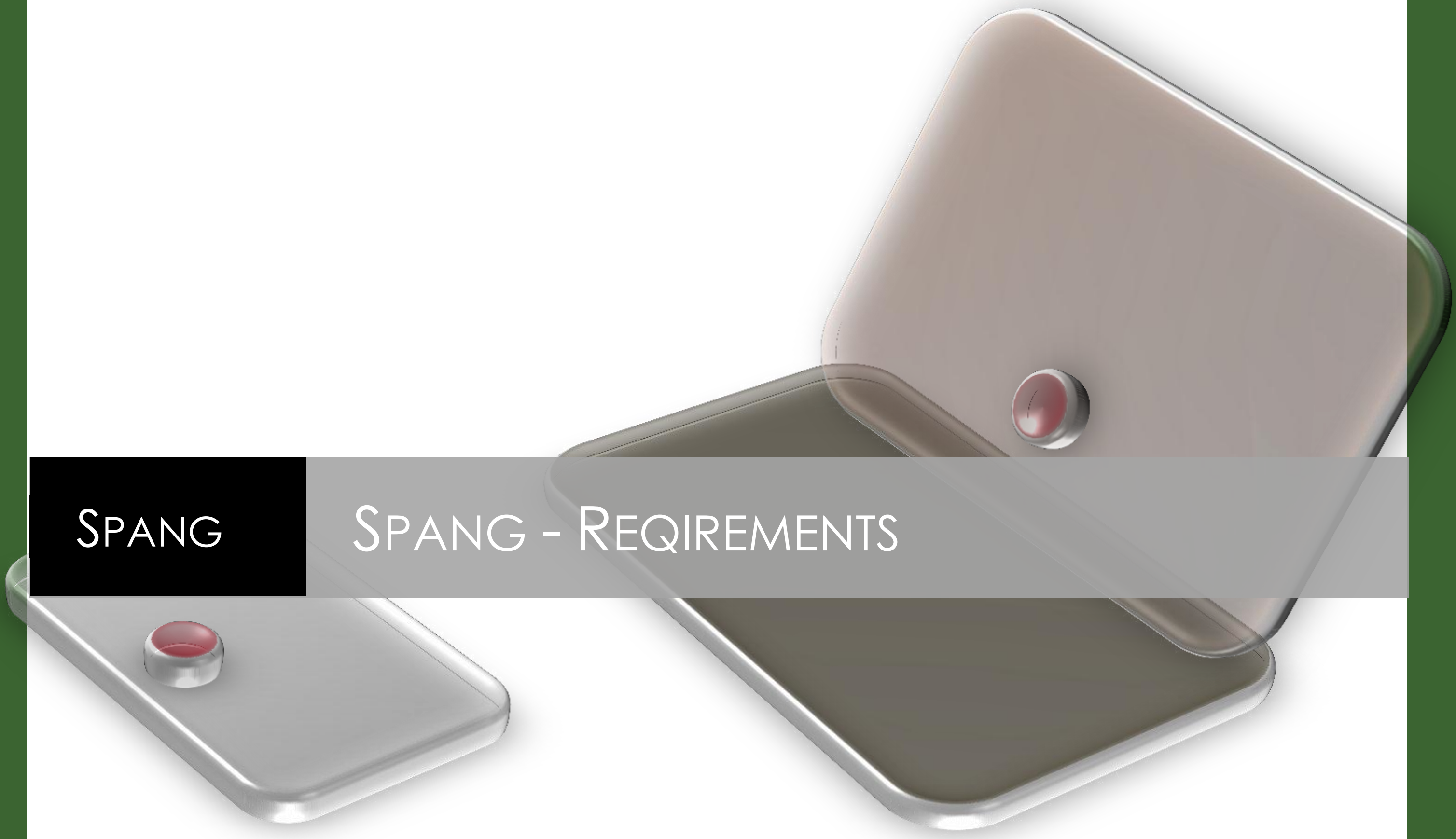
10/5/2012



Spang

Spang - Reqirements

Version 1.1 | Spang

TABLE OF CONTENTS

Functional requirements

Use the screen of the android device as a touch-pad

Since controlling the mouse might be desired even if the user sits at some distance from the computer, using the phone as a touch-pad would be a nice functionality. The mouse cursor is moved when the user moves a finger upon the screen of the android device.

Trigger:

- The user makes some kind of touch input on the android screen.

Precondition:

- The android device is connected to the server.

Basic path:

- The user chooses to control the mouse cursor using "touch" in the menu and is sent to the corresponding activity. As the user then touches the screen all the positions and the pressure of the active fingers are sent to the computer for interpretation.
  A circle is drawn on the android screen on each position of an active finger and follows the movements of the fingers. A single tap by default corresponds to a left-click, a long tap to a right-click and horizontal scrolling is triggered by a horizontal two finger swipe.

Exceptional path:

- The network connection is lost and the application tries to reconnect to the server. I this fails the user is sent back to the main menu.

Automatic reconnection to the network

If the app loses the connection to the computer it will automatically try to recreate the connection. Without this functionality the usage of the application on an unstable network would be irritating for the user, or in the worst scenario impossible.

Trigger:

- The existing connection to the network is lost.

Precondition:

- The android device has been, but is no more, connected to the network.

Basic path:

- Lukas

Exceptional path:

NEVER CRASH DUE TO NETWORK FAILURE

Our network code should be stable enough to not crash the program in case of eventualities like a lost network connection etc. If the program force closes every time the network connection is lost the user would be disappointed.

Trigger:

- The network isn't stable, e.g. loses connection between android and computer.

Precondition:

Basic path:

- The problem is handled without causing a force close.

Exceptional path:

- User is sent back to the start view?

SET THE UPDATE FREQUENCY FOR EACH SUPPORTED SENSOR.

The user should be able to choose how often the data from each sensor should be updated through the application preferences. This is available so that the user has the option to lessen the battery consummation of the application.

Trigger:

- The user sets the update frequency for an arbitrary sensor in the preferences view.

Precondition:

- The app is running.

Basic path:

- The user clicks the menu button on the device and chooses “settings”. A list of all sensors that supported in Spang *and* available in the device is shown. By clicking on the update frequency preference on any of the sensors the user is able to set the desired frequency.

Exceptional path:

- The sensor is not supported by Spang or not available on the device. The sensor is therefore not shown in the list and has no preferences to be edited.

Send and images from phone to computer.

The user should be able to send a stored or newly captured image from the phone to the computer. This would be very useful since its quite complicated to transfer images from phone to computer today. Using the Spang API we eliminate the need for using cloud services, physical cables etc.

Trigger:

- The user selects an image from a gallery or takes a new one using the camera and chooses to send it to the computer.

Precondition:

- There are stored images to send or the device is equipped with a camera.

Basic path:

- The user selects an image to send (or captures a new one with the camera) and the android application sends it to the computer.

Exceptional path:

- There is no network connection. The user is informed of this and can’t send any image.

Exceptional path: T

- here is camera. The user is informed of this and can’t send any image.

Create and use macros.

The user should be able to save sequences of pressed button combinations in order to execute them on the computer. These button presses will be easily accessible in the app through key presses etc. in order to execute commands easily from the phone. This is not something available in normal mobile keyboards.

Trigger:

- The user chooses to send a pre-stored macro through the app menu.

Precondition:

- There has to be macros pre-stored in the app

Basic path:

- The user chooses to send a created macro, e.g. “ctrl+alt-del”. The macro is then sent and interpreted by the computer software which sends the corresponding button presses to the computer OS.

Exceptional path:

- There are no stored macros. The user has the option to cancel or to create a new macro.

Exceptional path:

- There is no network connection. The user is informed of this and can’t send any macros.

Support for multiple android devices (needed in e.g. multiplayer games etc.)

Several android devices should be able to connect to the same computer and interact in the same computer application.

Trigger:

- An additional android device connects to the computer using Spang.

Precondition:

- When triggered there are at least two connected android devices.

Basic path:

- The computer application notifies than an additional device has connected. All android devices transfer and receive data to and from the computer.

Exceptional path:

- One of the devices loses connection. The device tries to reconnect.

Extended keyboard

A keyboard that allows the user to use buttons normally only found on computer keyboards (e.g. ctrl). Since the app should be able to control a computer all functions should be accessible.

Trigger:

- The user chooses keyboard on the input menu

Precondition:

- In order to send input, the android app must be connected to the computer

Basic path:

- An extended keyboard is launched.

Exceptional path:

- If the connection is lost, it will be handled following the application standard and ask if user want to reconnect.

GUI REQUIREMENTS

ABILITY TO SEND AND RECEIVE FROM ALL AVAILABLE SENSORS

The android application should be able to decide which supported sensors are available (and enabled) on the android device and send those. The computer application should be able to receive the sensor data and interpret in correctly, executing the correct commands. It is the intention of this API to enable users to develop their own computer applications making use of the sensor input from the phone(s).

WELL DOCUMENTED API/INTERFACE

All classes and public methods should have an informative Javadoc. Since those that would use the API will not usually be able to communicate directly with its creators, it is imperative that the documentation is good; since they have to entrust it in order to understand how to implement the API correctly.

API-WIKI

Since a Java Doc alone does not provide a clear path into the API, a wiki will also be provided for the different parts of Spang.

UNIT-TEST OF THE NETWORK COMMUNICATION

Provided that a part of the network code can be unit tested, it also should be unit tested. This is important since unit tests often are able to find bugs that wouldn't otherwise be found.

SAMPLe PROJECT

In order to helpfully show how the API is supposed to be used, a sample project will be available.