10/5/2012

|  |
| --- |
| Version 1.1 | Spang |

|  |  |
| --- | --- |
| Spang | Spang - Reqirements |

Table of Contents

[Functional requirements 2](#_Toc337732523)

[Use the screen of the android device as a touch-pad 2](#_Toc337732524)

[Automatic reconnection to the network 2](#_Toc337732525)

[Never crash due to network failure 3](#_Toc337732526)

[Set the update frequency for each supported sensor. 3](#_Toc337732527)

[Send and images from phone to computer. 4](#_Toc337732528)

[Create and use macros. 4](#_Toc337732529)

[Support for multiple android devices (needed in e.g. multiplayer games etc.) 5](#_Toc337732530)

[Extended keyboard 5](#_Toc337732531)

[GUI requirements 6](#_Toc337732532)

[Ability to send and receive from all available sensors 6](#_Toc337732533)

[Well documented API/interface 6](#_Toc337732534)

[API-wiki 6](#_Toc337732535)

[Unit-test of the network communication 6](#_Toc337732536)

[Sample project 6](#_Toc337732537)

# 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.

## 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.

## 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.

## 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.

## 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.

## 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.

## 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.

## 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.

# 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.