

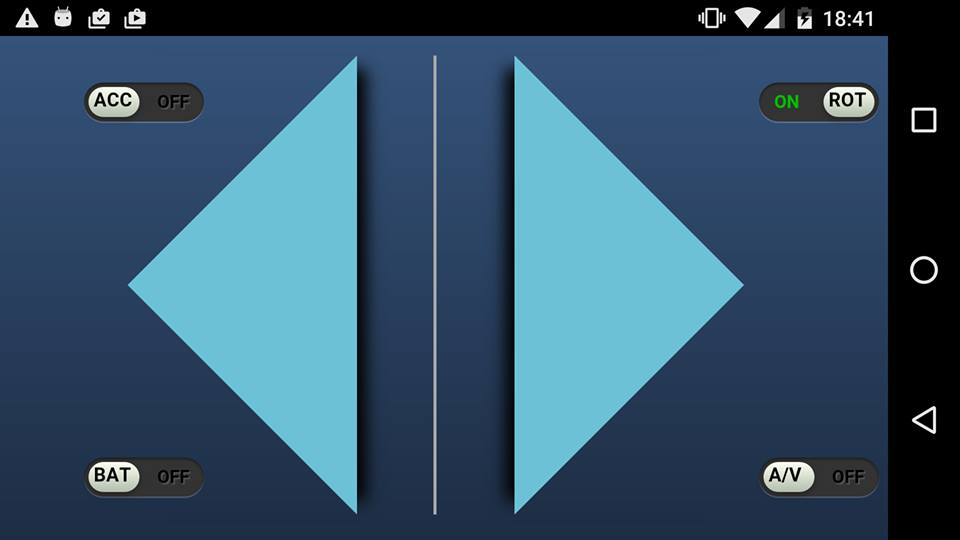
Smartphone Gamepad Framework

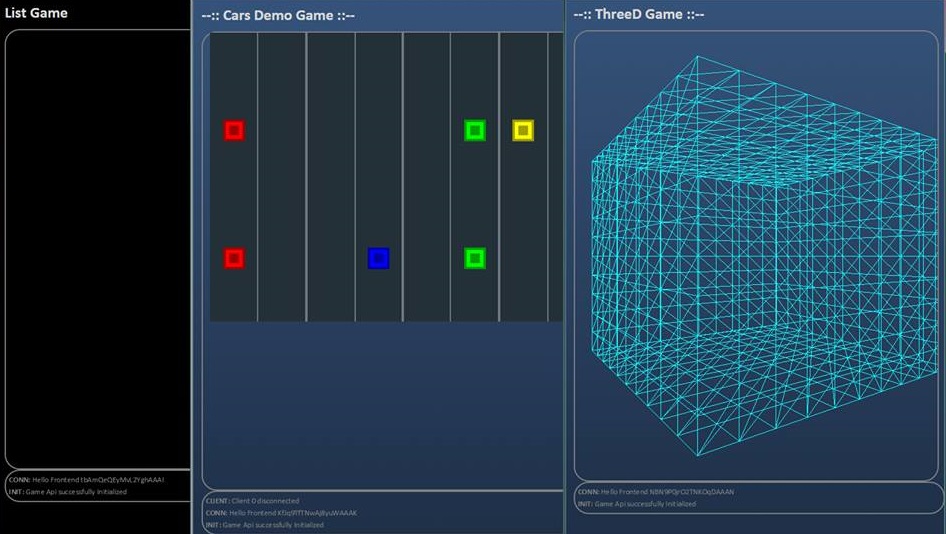
Ron Schiwkowski, Michael Schleiss, Dennis Hofmann, Christian Heinrichs, Hannes Grothkopf

Result and Outlook

The resulting application is able, after starting it within the browser, to **take the sensor data** from the smartphone and **use it for operations** on the desktop PC. This can be used for **presentations**, **games** and several other fields, as the **intuitive movement** makes every scenario imaginable.

Three example games and several templates for the gamepad were created:





Controller Template Example games

**Outlook**

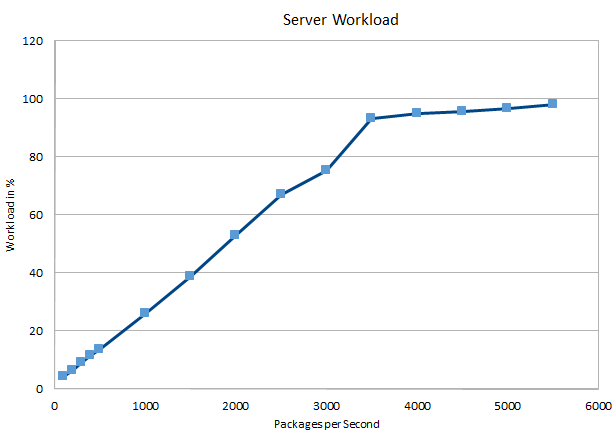
* Possibility to create an **App** out of the API and detach it from the browser
* Extend functionality to several available Sensors currently unused (light, humidity, …)
* Peaks in server workload

Data processing

To **lower the amount** of data sent over the network, the data is first **filtered**. For this, two steps are of great importance:

**Filtering of empty/unused data packages**

**Averaging of several packages to one**

These steps are necessary, as a currently normal interval of **50ms** and higher would overload the **UDP** connection used. 

Currently the application supports **70 devices** at an interval of 10 updates a second

Data extraction

Every average smartphone has a number of sensors, that continuously **observe the surrounding** environment. This data can be extracted from the device via **JavaScript**. For this purpose, some newly available **APIs** were used:

Rotation and Acceleration .   
The mostly used sensor data is the **movement in 3-dimensional space**. This information is collected via an intern gyroscope and accelerometer in a fixed interval **several times every second**. The easiest way to collect this is the **Eventlistener****'***devicemotion***'**, as shown in the following example:

**window**.addEventListener(**'devicemotion'**, **function** (event){  
**var** accX = event.acceleration.**x**; *//high, down* **var** rAlpha = event.rotationRate.**alpha**; *//rotation alpha*

Further Sensors in SCI: **Audio and Video** via getUserMedia() API

**Battery Information** via Battery Status API

**Orientation**

**Advantage**: The smartphone is a very intuitive device and in combination with the SGF usable in a way only modern high-prized consoles provide.

**Realization**: The sensor data is collected from the smartphone, filtered and processed, and finally sent via wireless connection to the server

commands

sensordata

**Goal**: Create an API, that allows an easy usage of any smartphone as a controller on every average computer.