Benjamin A.
Torsten K.

API - Kommunikation

Inhaltsverzeichnis

Änderungen zur letzten Version

- 22.04.2013 Funktion „text“ modifiziert, Text wird in 2 Zeilen gesplittet
- 09.05.2013 Funktion „tetris“ erstellt für Server und Client
- 09.05.2013 Typen für die Variablen dazu geschrieben, Formatierung der Inhalte für die Variablen geändert.
- 26.05.2013 Funktion „draw“ erstellt

API - Kommunikation

Auf- und Abbau einer Verbindung

Funktion „connect“

So bald eine Verbindung zwischen der Raspberry Pi und dem Smartphone besteht, wird vom Smartphone folgende Nachricht verschickt:

{
"function": "connect",
"status": "menu",
"udid": 1460542022
}
	wird vom **Client** verschickt

status - String	**menu** → App befindet sich im Menü **drawing** → App befindet sich im Mal-Modus **text** → App befindet sich im Text-Modus **tetris** → App befindet sich im Spiel-Modus (Tetris)
udid - Integer	Das Smartphone wird an Hand einer eindeutigen Nummer identifiziert, somit kann eine pausierte Session wieder aufgenommen werden.

In der Variable „status“ wird die aktive Activity vermerkt, somit weiß der Raspberry Pi in welchem Modus die Hard- und Software ist.

Der Server (Raspberry Pi) antwortet mit folgender Nachricht:

{
"function": "connect",
"status": "success"
}
	wird vom **Server** verschickt

status - String	**success** → Client kann nun die weiteren Funktionen nutzen **failure** → ein anderer Client ist eingeloggt

API - Kommunikation

Funktion „break“

So bald die Android App den Focus verliert (Bildschirm geht aus, andere App wird genutzt), wird die Verbindung pausiert. Dadurch werden die Ressourcen vom Smartphone freigegeben. Der Client kann innerhalb von 60 Sekunden die Verbindung mit der Funktion „connect“ wieder Aufbauen. Falls dies nicht passiert gibt der Server die reservierte Hardware frei und ein beliebiger Client kann sich auf die Hardware einloggen.

{
"function": "break"
}
	wird vom **Client** verschickt

Funktion „disconnect“

Der Client kann die Verbindung zum Server auch freigeben, falls der Client die Funktionen nicht mehr nutzen möchte. Dafür wird eine Nachricht mit der Funktion „disconnect“ verschickt.

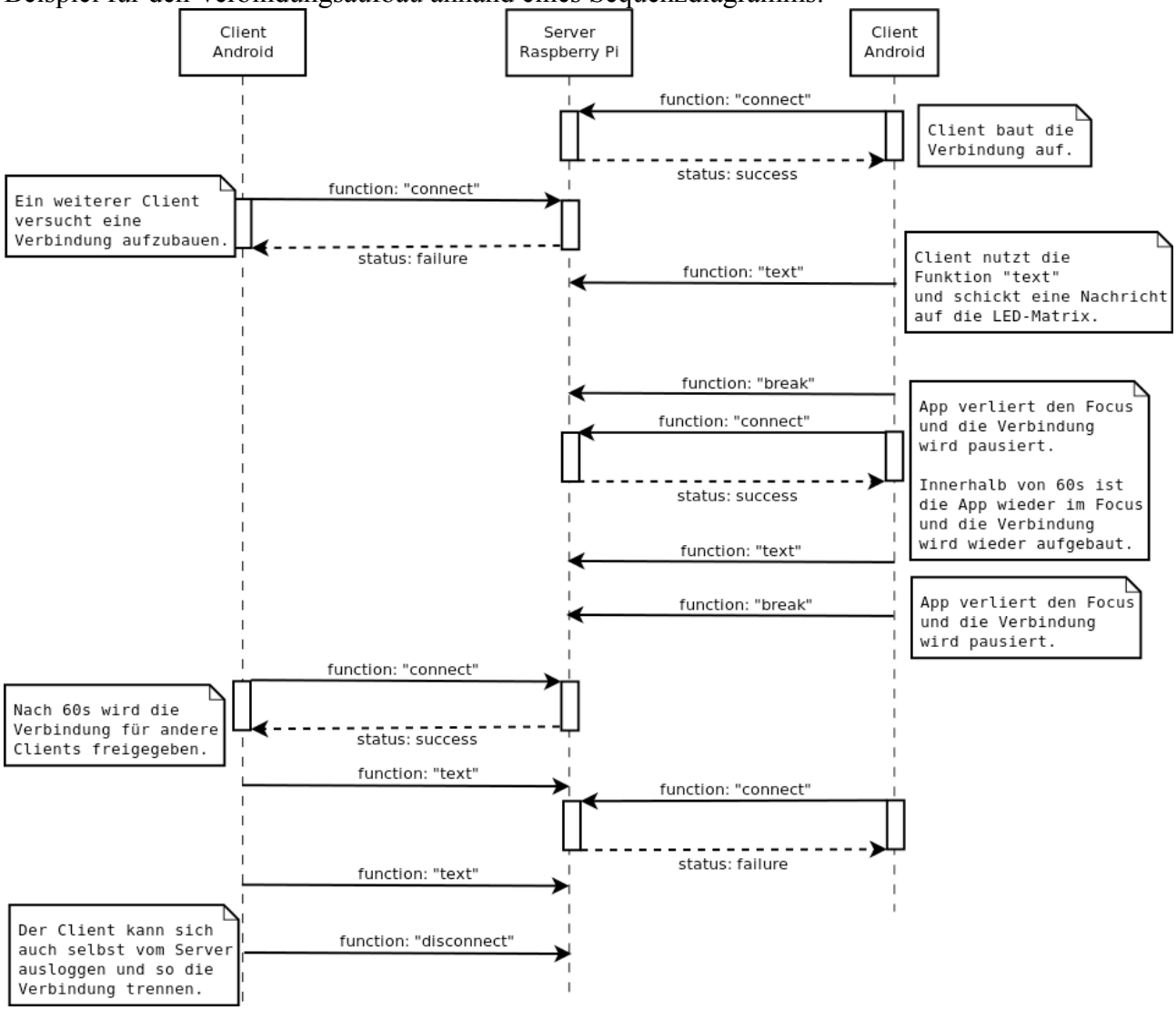
Aber auch der Server kann die Verbindung zum Client trennen. Dafür muss ein Taster an der LED-Wand betätigt werden und der Server schickt dann eine Nachricht mit der Funktion „disconnect“ an den Client.

{
"function": "disconnect"
}
	wird vom **Client** und vom **Server** verschickt

API - Kommunikation

Verbindungsaufbau Sequenzdiagramm

Beispiel für den Verbindungsaufbau anhand eines Sequenzdiagramms.



API - Kommunikation

Text an die LED Wall schicken

Nachrichten vom Client

Text senden – Funktion „text“	

{
"function": "text",
"text-top": "Zeile 1",
"text-bottom" : "Zeile 2",
"color":"red",
"inverted": false,
"animation": "none"
}
	Die Felder text-top oder text-bottom können auch einen leeren String enthalten. Ein Feld muss aber mindestens ein Zeichen enthalten

~~{~~ ~~"function": "text",~~ ~~"message": "meine Nachricht",~~ ~~"color":"red",~~ ~~"inverted": false,~~ ~~"animation": "none"~~ ~~}~~	**alte Version**: message wurde durch text-top und text-bottom ersetzt
text-top - String	stellt die erste Zeile dar
text-bottom - String	stellt die zweite Zeile dar
color - String	**red** → Text/Hintergrund in rot **green** → Text/Hintergrund in grün **blue** → Text/Hintergrund in blau **multicolored** → Text/Hintergrund in allen Farben
inverted - Boolean	**false** → Text ist in Farbe (Hintergrund aus) **true** → Hintergrund ist in Farbe (Text aus)
animation - String	**none** → keine Animation

API - Kommunikation

Zeichnen auf der LED Wall

Mit der Zeichen-Funktion kann man einzelne LEDs vom Smartphone aus steuern, dadurch hat man die Möglichkeiten eigene Bilder auf dem 10x16 Feld zu zeichnen.

Nachrichten vom Client

Tetris spielen – Funktion „tetris“	
{"function":"draw","data": [3,2,1,3,1,0,1,0,1,1,2,2,0,0,1,1,0,0,2 ,1,0,0,0,1,0,2,0,1,0,0,0,0,1,0,1,0,1,0 ,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0 ,0,0,0,2,1,0,0,0,1,0,0,0,0,1,0,1,1,0,0 ,2,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1 ,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0 ,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,1 ,0,0,0,0,0,0,0,3]}	
data – Integer Array	Das Feld enthält 160 Einträge und repräsentiert alle Pixel auf der LED Wall. Folgende Werte für einen Eintrag sind möglich: **0** → LED aus **1** → rot **2** → grün **3** → blau

API - Kommunikation

Tetris – spielen auf der LED Wall

Text....

Nachrichten vom Server

Tetris spielen – Funktion „tetris“	
{ "function": "tetris", "gameState": { "score": 0, "level": 1, "isRunning": true, "nextStone": "J" } }	
score - Integer	aktueller Punktestand
level - Integer	aktuelles Level
isRunning - Boolean	Gibt den Status vom Spiel an • **true** → Spiel läuft (es fallen Steine) • **false** → Spiel ist beendet (es können keine Steine mehr fallen)
nextStone - String	Der nächste Stein, welcher nach dem aktuellen Stein folgt, der fallen wird. Mögliche Steine: I, J, L, O, S, T, Z

API - Kommunikation

Nachrichten vom Client

Tetris spielen – Funktion „tetris“	

{
"function": "tetris",
"action": "init"
}	

action - String	stellt die nächste Aktion von der Activity bzw. vom Spieler dar: • **init** → User startet die Activitiy und der Server wird benachrichtigt • **start** → Der User hat in der Activity den Start Button betätigt und beginnt nun das Spiel • **quit** → Der User beendet das Spiel bevor es zu Ende ist (Wechsel der Activity innerhalb der App oder Stopp Taste in der Activity) In action werden auch die Bewegungen vom Stein aufgenommen: • **L** → Stein nach links bewegen • **R** → Stein nach rechts bewegen • **C** → Stein im Uhrzeigersinn drehen um 90° (nach rechts) • **CC** → Stein gegen den Uhrzeigersinn um 90° drehen (nach links)